

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

IN RE APPLICATION OF: Nobuaki OTSUKA

GAU:

SERIAL NO: New Application

EXAMINER:

FILED: Herewith

FOR: FUSE CIRCUIT

REQUEST FOR PRIORITY

COMMISSIONER FOR PATENTS
ALEXANDRIA, VIRGINIA 22313

SIR:

- ☐ Full benefit of the filing date of U.S. Application Serial Number _____, filed _____, is claimed pursuant to the provisions of 35 U.S.C. §120.
- ☐ Full benefit of the filing date(s) of U.S. Provisional Application(s) is claimed pursuant to the provisions of 35 U.S.C. §119(e):
Application No. _____ Date Filed _____
- ☒ Applicants claim any right to priority from any earlier filed applications to which they may be entitled pursuant to the provisions of 35 U.S.C. §119, as noted below.

In the matter of the above-identified application for patent, notice is hereby given that the applicants claim as priority:

<u>COUNTRY</u>	<u>APPLICATION NUMBER</u>	<u>MONTH/DAY/YEAR</u>
Japan	2003-290861	August 8, 2003

Certified copies of the corresponding Convention Application(s)

- ☒ are submitted herewith
- ☐ will be submitted prior to payment of the Final Fee
- ☐ were filed in prior application Serial No. _____ filed _____
- ☐ were submitted to the International Bureau in PCT Application Number _____
Receipt of the certified copies by the International Bureau in a timely manner under PCT Rule 17.1(a) has been acknowledged as evidenced by the attached PCT/IB/304.
- ☐ (A) Application Serial No.(s) were filed in prior application Serial No. _____ filed _____; and
- ☐ (B) Application Serial No.(s) _____
☐ are submitted herewith
- ☐ will be submitted prior to payment of the Final Fee

Respectfully Submitted,

OBLON, SPIVAK, McCLELLAND,
MAIER & NEUSTADT, P.C.



Marvin J. Spivak

Registration No. 24,913

Customer Number

22850

Tel. (703) 413-3000
Fax. (703) 413-2220
(OSMMN 05/03)

C. Irvin McClelland
Registration Number 21,124

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 3 年 8 月 8 日
Date of Application:

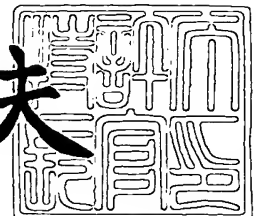
出 願 番 号 特 願 2 0 0 3 - 2 9 0 8 6 1
Application Number:
[ST. 10/C]: [J P 2 0 0 3 - 2 9 0 8 6 1]

出 願 人 株 式 会 社 東 芝
Applicant(s):

2 0 0 3 年 9 月 2 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



【書類名】 特許願
【整理番号】 A000302393
【提出日】 平成15年 8月 8日
【あて先】 特許庁長官 殿
【国際特許分類】 H01L 21/82
【発明者】
 【住所又は居所】 神奈川県川崎市幸区小向東芝町 1 番地 株式会社東芝マイクロエ
 レクトロニクスセンター内
 【氏名】 大塚 伸朗
【特許出願人】
 【識別番号】 000003078
 【氏名又は名称】 株式会社 東芝
【代理人】
 【識別番号】 100058479
 【弁理士】
 【氏名又は名称】 鈴江 武彦
 【電話番号】 03-3502-3181
【選任した代理人】
 【識別番号】 100091351
 【弁理士】
 【氏名又は名称】 河野 哲
【選任した代理人】
 【識別番号】 100088683
 【弁理士】
 【氏名又は名称】 中村 誠
【選任した代理人】
 【識別番号】 100108855
 【弁理士】
 【氏名又は名称】 蔵田 昌俊
【選任した代理人】
 【識別番号】 100084618
 【弁理士】
 【氏名又は名称】 村松 貞男
【選任した代理人】
 【識別番号】 100092196
 【弁理士】
 【氏名又は名称】 橋本 良郎
【手数料の表示】
 【予納台帳番号】 011567
 【納付金額】 21,000円
【提出物件の目録】
 【物件名】 特許請求の範囲 1
 【物件名】 明細書 1
 【物件名】 図面 1
 【物件名】 要約書 1

【書類名】 特許請求の範囲**【請求項 1】**

複数のブロックから構成されるフューズセットを具備し、前記複数のブロックの各々は、電氣的にプログラム可能な複数のフューズ素子から構成されるサブフューズセットと、前記複数のフューズ素子に対するプログラムを制御するプログラム制御回路とを有し、前記複数のフューズ素子のうちの 1 つは、前記サブフューズセットの有効／無効を示すイネーブルビットであり、前記プログラム制御回路は、前記イネーブルビットの値に基づいて、前記複数のブロックのなかから前記プログラムの対象となるブロックを決定することを特徴とするフューズ回路。

【請求項 2】

請求項 1 記載のフューズ回路において、さらに、前記プログラムの対象となるブロックに対するプログラムデータをラッチするプログラムデータラッチ回路を具備することを特徴とするフューズ回路。

【請求項 3】

前記複数のブロックは、 n (n は、複数) 段に直列に接続され、1 段目のブロックでは、前記プログラム制御回路は、前記 1 段目のブロック内の前記イネーブルビットの値に基づいて、前記 1 段目のブロックが前記プログラムの対象となるか否かを決定することを特徴とする請求項 1 記載のフューズ回路。

【請求項 4】

前記 1 段目のブロック内の前記イネーブルビットの値が前記サブフューズセットの無効を示しているとき、前記 1 段目のブロックは、前記プログラムの対象になることを特徴とする請求項 3 記載のフューズ回路。

【請求項 5】

前記複数のブロックは、 n (n は、複数) 段に直列に接続され、 i ($2 \leq i \leq n$) 段目のブロックでは、前記プログラム制御回路は、($i - 1$) 段目のブロック内の前記イネーブルビットの値及び前記 i 段目のブロック内の前記イネーブルビットの値に基づいて、前記 i 段目のブロックが前記プログラムの対象となるか否かを決定することを特徴とする請求項 1 記載のフューズ回路。

【請求項 6】

前記 ($i - 1$) 段目のブロック内の前記イネーブルビットの値が前記サブフューズセットの有効を示し、前記 i 段目のブロック内の前記イネーブルビットの値が前記サブフューズセットの無効を示しているとき、前記 i 段目のブロックは、前記プログラムの対象になることを特徴とする請求項 5 記載のフューズ回路。

【請求項 7】

前記複数のブロックの各々は、前記複数のフューズ素子にプログラムされたフューズデータのリードを制御するリード制御回路を有し、前記リード制御回路は、前記イネーブルビットの値に基づいて、前記複数のブロックのなかから前記リードの対象となるブロックを決定することを特徴とする請求項 1 記載のフューズ回路。

【請求項 8】

前記複数のブロックは、 n (n は、複数) 段に直列に接続され、 i ($1 \leq i \leq n - 1$) 段目のブロックでは、前記リード制御回路は、前記 i 段目のブロック内の前記イネーブルビットの値及び ($i + 1$) 段目のブロック内の前記イネーブルビットの値に基づいて、前記 i 段目のブロックが前記リードの対象となるか否かを決定することを特徴とする請求項 7 記載のフューズ回路。

【請求項 9】

前記 i 段目のブロック内の前記イネーブルビットの値が前記サブフューズセットの有効を示し、前記 ($i + 1$) 段目のブロック内の前記イネーブルビットの値が前記サブフューズセットの無効を示しているとき、前記 i 段目のブロックは、前記リードの対象になることを特徴とする請求項 8 記載のフューズ回路。

【請求項 10】

前記複数のブロックは、 n (n は、複数) 段に直列に接続され、 n 段目のブロックでは、前記リード制御回路は、前記 n 段目のブロック内の前記イネーブルビットの値に基づいて、前記 n 段目のブロックが前記リードの対象となるか否かを決定することを特徴とする請求項 7 記載のフューズ回路。

【請求項 11】

前記 n 段目のブロック内の前記イネーブルビットの値が前記サブフューズセットの有効を示しているとき、前記 n 段目のブロックは、前記リードの対象になることを特徴とする請求項 10 記載のフューズ回路。

【請求項 12】

前記複数のブロックの各々は、前記複数のフューズ素子からリードされたフューズデータをラッチするリードデータラッチ回路を有することを特徴とする請求項 1 記載のフューズ回路。

【請求項 13】

前記複数のブロックの各々は、前記複数のフューズ素子のうちの 1 つからリードされた前記イネーブルビットをラッチするリードデータラッチ回路を有し、かつ、前記イネーブルビットを除く、残りのフューズデータをラッチするリードデータラッチ回路は、前記複数のブロックに共有されることを特徴とする請求項 1 記載のフューズ回路。

【請求項 14】

前記複数のフューズ素子の各々は、電氣的フューズ素子又はアンチフューズ素子であることを特徴とする請求項 1 記載のフューズ回路。

【請求項 15】

前記プログラムは、パッケージング工程前又は後の任意の時期に実行されることを特徴とする請求項 1 記載のフューズ回路。

【請求項 16】

前記フューズセットに対するデータ書き換え回数の上限は、前記複数のブロックの数に等しいことを特徴とする請求項 1 記載のフューズ回路。

【請求項 17】

前記複数のブロックは、 n (n は、複数) 段に直列に接続され、データ書き換えの度に、前記プログラムの対象となるブロックは、1 段目のブロックから n 段目のブロックに向かって移動していくことを特徴とする請求項 1 記載のフューズ回路。

【請求項 18】

1 つ又は複数の機能ブロックを有する集積回路において、前記集積回路に関するデータを不揮発に記憶するメモリを具備し、前記メモリは、請求項 1 記載のフューズ回路であり、前記集積回路に関するデータは、書き換え可能であることを特徴とする集積回路。

【請求項 19】

n (n は、複数) 段に直列に接続された複数のサブフューズセットを有するフューズ回路に対して、プログラムの度に、前記プログラムの対象となるサブフューズセットを、1 段目のサブフューズセットから n 段目のサブフューズセットに向かって、順次、自動的に変えていくことにより、前記フューズ回路に対するデータ書き換えを可能にすることを特徴とするリード／プログラム方法。

【請求項 20】

前記プログラムの対象となるサブフューズセットは、パワーオンの直後に、前記サブフューズセットの有効／無効を示すイネーブルビットの値に基づいて自動的に決定されることを特徴とする請求項 19 記載のリード／プログラム方法。

【請求項 21】

前記パワーオンの直後に、前記イネーブルビットの値に基づいて、フューズデータリードの対象となるサブフューズセットが自動的に決定されることを特徴とする請求項 20 記載のリード／プログラム方法。

【請求項 22】

前記フューズデータリードの対象となるサブフューズセットは、最新のデータを記憶し

ていることを特徴とする請求項 2 1 記載のリード／プログラム方法。

【請求項 2 3】

前記パワーオンの直後に、前記イネーブルビットと共に全てのフューズデータがリードデータラッチ回路にラッチされることを特徴とする請求項 2 1 記載のリード／プログラム方法。

【請求項 2 4】

前記パワーオンの直後に、まず、前記イネーブルビットがリードデータラッチ回路にラッチされ、前記フューズデータリードの対象となるサブフューズセットが決定された後に、そのサブフューズセットのフューズデータが前記リードデータラッチ回路にラッチされることを特徴とする請求項 2 1 記載のリード／プログラム方法。

【書類名】明細書

【発明の名称】フューズ回路

【技術分野】

【0001】

本発明は、標準CMOSプロセスで形成できる電氣的にプログラム可能なフューズ素子に対するリード／プログラム制御回路に関し、特に、電氣的(electrically)-フューズ素子、アンチフューズ素子などから構成されるフューズ回路に使用される。

【背景技術】

【0002】

従来、半導体集積回路内においてデータを不揮発に記憶しようとした場合、その半導体集積回路内に何らかしらのメモリ素子を配置しなければならない。

【0003】

例えば、記憶するデータのデータ量が非常に多く、かつ、データの書き換えが多数回行われるような場合には、スタックトゲート構造のフラッシュメモリセルによりそのデータを記憶する。ここで、フラッシュメモリセルは、標準CMOSプロセスとは異なる特殊プロセスにより形成される。このため、ロジックLSIなどの標準CMOSプロセスで形成される半導体集積回路内にフラッシュメモリセルを配置する場合には、メモリ容量の大容量化を図るなど、特殊プロセスを用いることによるコストオーバーヘッドを抑える工夫が必要になる。

【0004】

しかし、例えば、1チップ内に複数の機能を混載させたシステムLSIなどの半導体集積回路においては、データ量の多いデータを不揮発に記憶する、という要求がなされることは、ほとんどない。このような半導体集積回路では、回路動作に関するトリミングデータ、チップID、セキュリティデータ、DRAMやSRAMなどのリダンダンシイデータなどの少ないデータ量のデータを不揮発に記憶するための小容量のメモリ素子が存在していれば足りる。

【0005】

また、このようなデータは、1回、又は、多くても数回の書き換えが行えれば、それで十分であり、フラッシュメモリセルのような10万回といったデータ書き換え回数を保障する必要は全くない。

【0006】

従って、システムLSIなどの半導体集積回路においては、特殊プロセスではなく、標準CMOSプロセスで形成することができる不揮発性メモリセルによりこのようなデータを記憶し、コストの低下を図ることが重要となる。

【0007】

ところで、標準CMOSプロセスで形成できる不揮発性メモリセルとしては、フューズ素子がある(例えば、特許文献1, 2参照)。

【0008】

フューズ素子と言えば、従来、メタル配線やポリシリコン配線をレーザにより切断するレーザ溶断タイプフューズ素子がよく知られていたが、現在では、パッケージング後にもプログラムを可能にする、という要望から、このようなレーザ溶断タイプフューズに代わって、電氣的にプログラム可能なフューズ素子(E(electrical)-fuse、Anti-fuse など)を用いることが主流となりつつある。

【0009】

電氣的にプログラム可能なフューズ素子としては、例えば、過大電流による配線の溶断によりプログラムを行うタイプのものや、電圧ストレスによる絶縁体の破壊によりプログラムを行うタイプのものなどがある。前者のフューズ素子では、配線の溶断により導通状態から非導通状態になるが、後者のフューズ素子では、絶縁体の破壊により非導通状態から導通状態になる。

【0010】

また、前者のフューズ素子と後者のフューズ素子の中間的なものとして、サリサイド層とポリシリコン層とからなる配線をフューズ素子として用いたものがある。このフューズ素子では、過大電流によるサリサイド層の破壊（抵抗値の増大）によりプログラムを行い、サリサイド層の破壊の前後の抵抗比を用いてデータを読み出す。

【0011】

図18は、従来のフューズ回路（フューズセット1個分）の例を示している。

【0012】

このフューズ回路は、1個のイネーブルビットフューズ素子及びm個のデータフューズ素子からなるフューズセットFSと、これら(m+1)個のフューズ素子にプログラムされるデータ E_n , D_{in1} , D_{in2} , ..., D_{inm} をラッチするプログラムデータラッチ回路PDLと、これら(m+1)個のフューズ素子から読み出されるデータ E_n , D_{out1} , D_{out2} , ..., D_{outm} をラッチするリードデータ（フューズデータ）ラッチ回路RDLとから構成される。

【0013】

イネーブルビットデータ E_n 及びプログラムデータ D_{in1} , D_{in2} , ..., D_{inm} は、このフューズ回路が形成されるチップの内部で生成されてもよいし、そのチップの外部から供給されてもよい。また、イネーブルビットデータ E_n 及びフューズデータ D_{out1} , D_{out2} , ..., D_{outm} は、例えば、パワーオン時に、リードデータラッチ回路RDLにラッチされた後、チップ内の回路に供給される。

【0014】

イネーブルビットデータ E_n は、m個のデータフューズ素子に記憶されたデータの有効／無効を判断するために使用される。例えば、イネーブルビットデータ E_n がプログラムされた状態、例えば、“1”（＝“H”）ならば、m個のデータフューズ素子に記憶されたデータは有効となるが、イネーブルビットデータ E_n がプログラムされていない状態、例えば、“0”（＝“L”）ならば、m個のデータフューズ素子に記憶されたデータは、無効となる。

【0015】

図19は、図18のフューズ回路の一部を具体的に示している。

本例では、フューズ素子FUSEとして、絶縁体の破壊によりプログラムを行うアンチフューズ素子を使用する。

【0016】

プログラム前においては、フューズセットFS内の全てのフューズ素子FUSEは、非導通状態となっている（初期状態）。ここで、プログラムデータ D_{in1} , D_{in2} , ..., D_{inm} の値に応じて、フューズ素子FUSEに対するプログラミングを実行する。

【0017】

プログラム時、プログラムの対象となるフューズセットにおけるイネーブルビットデータ E_n は、“1”となり、プログラムデータラッチ回路PDL（ E_n ）にラッチされる。この時、プログラムデータラッチ回路PDL（ E_n ）の出力は、“L”、例えば、接地電位になる。選択信号SELが“H”になると、トランジスタN2がオンし、フューズセットFS（ E_n ）内のフューズ素子FUSEの両端に高電圧VPPが印加される。

【0018】

その結果、フューズ素子FUSEにデータ“1”がプログラムされる。即ち、フューズ素子FUSEを構成する絶縁体が破壊され、フューズ素子FUSEは、非導通状態から導通状態に変化する。

【0019】

また、プログラム時、プログラムデータ D_{in1} , D_{in2} , ..., D_{inm} は、“1”又は“0”になる。例えば、プログラムデータ D_{in1} が“1”のときは、イネーブルビットデータ E_n のときと同様に、フューズ素子FUSEにデータ“1”がプログラムされる。プログラムデータ D_{in1} が“0”のときは、フューズセットFS（ D_{in1} ）内のフューズ素子FUSEの両端に高電圧VPPが印加されない。このため、フューズ素子

FUSEを構成する絶縁体が破壊されず、フューズ素子FUSEには、“0”がプログラムされる。

【0020】

実際に、フューズデータを使用するためには、フューズ素子FUSEに記憶されたアナログデータをデジタルデータに変換しなければならない。そこで、例えば、パワーオン時に、フューズデータを読み出し、これをリードデータ（フューズデータ）ラッチ回路RDLにラッチする。

【0021】

具体的には、まず、パワーオン時、リセット信号RSTが一時的に“H”になり、ラッチ回路の状態がリセットされる、即ち、出力データEn, Dout1, Dout2, …, Doutmは、全て、“L”となる。この後、フューズ素子FUSEのデータがリードデータラッチ回路RDLに転送される。

【0022】

例えば、フューズセットFS（En）内のフューズ素子FUSEが破壊状態（フューズデータ“1”）にあるときは、リードデータラッチ回路RDLには、“1”がラッチされ、その出力データEnは、“H”になる。また、フューズセットFS（En）内のフューズ素子FUSEが非破壊状態（フューズデータ“0”）にあるときは、リードデータラッチ回路RDLには、“0”がラッチされ、その出力データEnは、“L”になる。

【0023】

そして、電源が入っている間は、リードデータラッチ回路RDLは、常に、フューズデータをラッチしており、このフューズデータは、出力データDout1, Dout2, …, Doutmとして、データバスを経由して、内部回路やチップの外部に出力される。

【特許文献1】特開2002-76126号公報

【特許文献2】特開2002-368096号公報

【発明の開示】

【発明が解決しようとする課題】

【0024】

このように、電氣的にプログラム可能なフューズ素子によれば、パッケージング後にもフューズプログラムを行える、という利点を有するが、プログラム自体は、配線の溶断、絶縁体又はサリサイド層の破壊など、部材の物理的な変化に基づいており、一度、プログラムを行うと、プログラム前の元の状態に戻すことはできない。

【0025】

つまり、従来のフューズ回路では、1つのフューズセットに対するプログラミングは、1回のみであり、同一のフューズセットに対してフューズデータの書き換えを行う、ということは、不可能である。

【0026】

しかし、近年の半導体集積回路においては、少数回ではあるが、フューズデータの書き換えを行いたい、という要望がある。

【0027】

この要望を満たす一つの案としては、フラッシュメモリセルを搭載する、というものがあるが、上述のように、特殊プロセスが必要となり、コストの増大を招くため、採用することはできない。また、他の案としては、複数のフューズセットを外部信号により選択する機能を設けてフューズデータの書き換えを可能とする、というものも考えられるが、この場合、回路規模の増大や複雑化を招く。

【0028】

本発明の目的は、標準CMOSプロセスで形成でき、かつ、1つのフューズセットに対して、フューズデータの書き換えを、特殊な制御なしに、自動的に行うことができるフューズ回路を提案することにある。

【課題を解決するための手段】

【0029】

本発明の例に関わるフューズ回路は、複数のブロックから構成されるフューズセットを備え、複数のブロックの各々は、電氣的にプログラム可能な複数のフューズ素子から構成されるサブフューズセットと、複数のフューズ素子に対するプログラムを制御するプログラム制御回路とを有し、複数のフューズ素子のうちの1つは、サブフューズセットの有効／無効を示すイネーブルビットであり、プログラム制御回路は、イネーブルビットの値に基づいて、複数のブロックのなかからプログラムの対象となるブロックを決定する。

【発明の効果】

【0030】

本発明の例によれば、標準CMOSプロセスで形成でき、かつ、1つのフューズセットに対して、フューズデータの書き換えを、特殊な制御なしに、自動的に行うことができるフューズ回路を提供することができる。

【発明を実施するための最良の形態】

【0031】

以下、図面を参照しながら、本発明の例を実施するための最良の形態について詳細に説明する。

【0032】

1. 概要

小さな容量のデータを不揮発に記憶するためのメモリを、標準CMOSプロセスにより形成しようとする場合、そのメモリは、フューズ素子（例えば、MOSトランジスタのゲート構造を利用したアンチフューズなど）となる。しかし、フューズ素子に対するデータプログラムは、部材の溶断や破壊などにより物理的に行われるため、フューズ素子自体を元の状態に戻すことはできない。

【0033】

そこで、本発明の例では、フューズデータの書き換えを行うために、1つのフューズセットを複数のフューズブロックから構成する。各フューズブロック内には、複数のフューズ素子からなるサブフューズセットが配置され、このサブフューズセットにフューズデータがプログラムされる。本発明の例に関わるフューズ回路よれば、1つのフューズセットに対するデータの書き換え回数の最大値は、その1つのフューズセット内のフューズブロックの数に等しくなる。

【0034】

フューズデータのプログラム時には、プログラム制御回路により、複数のフューズブロックのうちから1つのフューズブロックが選択される。その選択は、各フューズブロック内のイネーブルビットデータの値に基づいて行われる。イネーブルビットデータは、フューズデータをプログラミングするときに、これと同時にプログラミングされるため、複数のフューズブロックを直列接続することにより、データ書き換えに応じて、自動的に、プログラミングの対象となるフューズブロックを選択できる。

【0035】

フューズデータのリード時には、最も新しいフューズデータがプログラムされているフューズブロックからフューズデータを、リードし、かつ、ラッチ回路にラッチする必要がある。つまり、データ書き換えを行った場合には、古いフューズデータがリードされないように、それを無効としなければならない。そこで、リード制御回路により、複数のフューズブロックのうちから、その最も新しいフューズデータがプログラムされている1つのフューズブロックを選択する。その選択は、各フューズブロック内のイネーブルビットデータの値に基づいて、容易に行うことができる。

【0036】

このように、本発明の例に関わるフューズ回路では、1つのフューズセットを複数のフューズブロック（複数のサブフューズセット）から構成する。このような場合においても、プログラムデータとしては、小容量で、かつ、書き換え回数が少数回で足りるのものを対象とするため、面積的なペナルティは、それほど大きくならない。逆に、このような構成を採用することで、特殊プロセスを採用することなく、標準CMOSプロセスのみで、

半導体集積回路を形成できるため、安価で、適用範囲の広いフューズ回路を提供できる。

【0037】

また、フューズデータのプログラム／リードに関しては、プログラム／リード制御回路を用いて、特定のフューズブロックに対して自動的に行うため、チップ外部から見た場合には、本発明の例に関わるフューズ回路は、従来の書き換えを行うことができないフューズ回路に対して何ら変わるところはない。

【0038】

2. フューズ回路

図1は、フューズ回路の概要を示している。

【0039】

フューズ回路1は、内部回路の動作に関するトリミングデータなど、小容量のデータを不揮発に記憶するため、LSIチップ上の任意の領域に配置される。

【0040】

イネーブルビットデータ E_n 、プログラムデータ D_{in1} , D_{in2} , \dots , D_{inm} は、内部回路又はLSIチップの外部から供給され、プログラムデータラッチ回路PDLに一時的にラッチされる。プログラムデータラッチ回路PDLは、例えば、1個又は複数個、本例では、 k 個のフューズセットに対して1つのみ設けられる。

【0041】

フューズ回路1内には、 k 個のフューズセットが配置される。本例では、1個のフューズセットと1個のリードデータラッチ回路とは、ペアとなっており、1つのフューズセットからフューズデータ D_{out1} , D_{out2} , \dots , D_{outm} が出力される。

【0042】

データのプログラム時には、例えば、選択信号 $SEL1$, $SEL2$, \dots , $SELk$ により1つのフューズセットが選択され、この選択されたフューズセットに対して、データのプログラミングが実行される。

【0043】

ここで、従来のフューズ回路では、1つのフューズセットに対しては、一度、データをプログラミングしたら、その後、そのデータを書き換える、ということはできない。これに対し、本発明の例に関わるフューズ回路では、1つのフューズセットに対しては、予め決められた所定の回数だけ、データを書き換えを行うことができる。

【0044】

このようなデータを書き換えは、後述するように、1つのフューズセット内に配置される複数個のサブフューズセットにより実現される。つまり、データ書き換えの度に、複数個のサブフューズセットのうちの1つにプログラムデータをプログラムし、かつ、最新のプログラムデータがプログラムされたサブフューズセットから常にデータが読み出されるようにする。

【0045】

なお、データ書き換え回数の上限は、1つのフューズセット内に配置されるサブフューズセットの数に等しくなる。

【0046】

以下、このようなデータ書き換えが可能なフューズ回路の具体例について説明する。

【0047】

3. 第1実施例

(1) フューズセット

図2は、本発明の第1実施例に関わるフューズ回路の主要部を示している。

【0048】

同図に示す構成は、図1の1つのフューズセットに対応している。1つのフューズセットは、 n (n は、複数) 個のフューズブロック $BLK1$, $BLK2$, \dots , $BLKn$ を有する。また、各フューズブロック $BLKi$ ($i=1, 2, \dots, n$) は、 $(m+1)$ 個のフューズ素子 (m は、複数) から構成されるサブフューズセット $SFSi$ と、このサブフ

ユーザセット SFS_i に対応して設けられる $(m+1)$ 個のプログラム制御回路 $PCNT_i$ 及び $(m+1)$ 個のリードデータラッチ回路 RDL_i とを有する。

【0049】

また、各フューズブロック BLK_i は、リード制御回路 $RCNT_i$ を有している。リード制御回路 $RCNT_i$ は、イネーブルビットデータを除く、 m ビットのフューズデータをラッチする m 個のリードデータラッチ回路 RDL_i に対応して、 m 個だけ設けられる。

【0050】

プログラムデータラッチ回路 PDL は、例えば、 n 個のフューズブロック BLK_1 , BLK_2 , \dots , BLK_n に対して1つのみ設けられる。プログラムデータラッチ回路 PDL は、イネーブルビットデータ E_n 及びプログラムデータ D_{in1} , D_{in2} , \dots , D_{inm} を一時的にラッチする。これら入力データ E_n , D_{in1} , D_{in2} , \dots , D_{inm} は、選択された1つのフューズブロック BLK_i に転送される。

【0051】

ここで、 n 個のフューズブロック BLK_1 , BLK_2 , \dots , BLK_n は、直列接続される。そして、プログラム制御回路 $PCNT_1$, $PCNT_2$, \dots , $PCNT_n$ は、イネーブルビットデータに基づいて、 n 個のフューズブロック BLK_1 , BLK_2 , \dots , BLK_n のうちから、プログラミングの対象となる1つのフューズブロック BLK_j を選択する。

【0052】

即ち、 i ($i=1, 2, \dots, n$) 番目のフューズブロック BLK_i では、プログラム制御回路 $PCNT_i$ は、フューズブロック BLK_i 内のイネーブルビットデータ E_{ni} の値と、1つ前の $(i-1)$ 番目のフューズブロック $BLK_{(i-1)}$ 内のイネーブルビットデータ $E_{n(i-1)}$ の値とに基づいて、データプログラムに関し、フューズブロック BLK_i の選択／非選択を決定する。

【0053】

但し、最初(1番目)のフューズブロック BLK_1 では、1つ前のフューズブロックが存在しない。また、最初にプログラムの対象となるフューズブロックは、フューズブロック BLK_1 である。そこで、最初のフューズブロック BLK_1 では、プログラム制御回路 $PCNT_1$ は、フューズブロック BLK_1 内のイネーブルビットデータ E_{n1} の値と、入力データ E_{n0} の値(“H” 固定)とに基づいて、フューズブロック BLK_1 の選択／非選択を決定する。

【0054】

なお、各フューズブロック BLK_i 内のイネーブルビットデータ E_{ni} は、例えば、パワーオンの後、直ちに、リードラッチ回路 RDL_i にラッチされ、イネーブルビットデータ E_{ni} として、各フューズブロック BLK_i にフィードバックされる。

【0055】

また、リード制御回路 $RCNT_1$, $RCNT_2$, \dots , $RCNT_n$ は、イネーブルビットデータに基づいて、 n 個のフューズブロック BLK_1 , BLK_2 , \dots , BLK_n のうちから、データリードの対象となる1つのフューズブロック BLK_j を選択する。

【0056】

即ち、 i ($i=1, 2, \dots, n$) 番目のフューズブロック BLK_i では、リード制御回路 $RCNT_i$ は、フューズブロック BLK_i 内のイネーブルビットデータ E_{ni} の値と、1つ後の $(i+1)$ 番目のフューズブロック $BLK_{(i+1)}$ 内のイネーブルビットデータ $E_{n(i+1)}$ の値とに基づいて、データリードに関し、フューズブロック BLK_i の選択／非選択を決定する。

【0057】

但し、最後(n 番目)のフューズブロック BLK_n では、1つ後のフューズブロックが存在しない。また、フューズブロック BLK_n は、最後にプログラムの対象となるフューズブロックである。そこで、最後のフューズブロック BLK_n では、リード制御回路 $RCNT_n$ は、フューズブロック BLK_n 内のイネーブルビットデータ E_{nn} の値と、入力デ

ータ E_n ($n+1$) の値 (“L” 固定) とに基づいて、フューズブロック BLK_n の選択／非選択を決定する。

【0058】

(2) 回路例

図3乃至図5は、図2のフューズセットを構成するブロック BLK_i の回路例を示している。

【0059】

[1] ブロック BLK_1

図3は、ブロック BLK_1 の回路例である。

1. プログラムデータラッチ回路

プログラムデータラッチ回路 $PDL(E_n)$ は、イネーブルビットデータ E_n をラッチする。プログラムデータラッチ回路 $PDL(D_{in1})$, \dots $PDL(D_{inm})$ は、 m ビットのプログラムデータ D_{in1} , \dots D_{inm} に対応して m 個だけ設けられ、かつ、プログラムデータ D_{in1} , \dots D_{inm} をラッチする。

【0060】

プログラムデータラッチ回路 $PDL(E_n)$, $PDL(D_{in1})$, \dots $PDL(D_{inm})$ は、全て、同じ回路構成を有している。

【0061】

インバータ I_1 及びクロックドインバータ CI_1 により、ラッチ回路が構成される。ラッチ回路の出力信号は、ノア回路 NR_1 の一方の入力端子に入力される。ノア回路 NR_1 の他方の入力端子には、プログラム信号 $PROG$ の反転信号 $bPROG$ が入力される。

【0062】

プログラム時には、プログラム信号 $PROG$ が “H” となり、かつ、その反転信号が “L” となる。このため、プログラム時には、ノア回路 NR_1 は、ラッチ回路にラッチされたデータに応じた出力信号を出力する。

【0063】

例えば、ラッチ回路にラッチされたデータが “1”、即ち、クロックドインバータ CI_1 の出力信号が “L” のときは、ノア回路 NR_1 の出力信号は、“H” となる。その結果、 N チャンネル MOS トランジスタ N_1 は、オンになる。また、ラッチ回路にラッチされたデータが “0”、即ち、クロックドインバータ CI_1 の出力信号が “H” のときは、ノア回路 NR_1 の出力信号は、“L” となる。その結果、 N チャンネル MOS トランジスタ N_1 は、オフになる。

【0064】

2. プログラム制御回路

プログラム制御回路 $PCNT_1$ は、アンド回路 AD_1 とプログラムスイッチ PSW とから構成される。

【0065】

アンド回路 AD_1 には、フューズセットの選択信号 SEL_1 、イネーブルビットデータ E_{n0} (“H” 固定) 及びイネーブルビットデータ E_{n1} の反転信号が、それぞれ入力される。フューズセットの選択信号 SEL_1 は、例えば、図1及び図2に示すように、フューズセット1に対するプログラミングを実行するときに、“H” となる。イネーブルビットデータ E_{n1} は、サブフューズセット SFS_1 に対するプログラミング前においては、“L”、即ち、リードデータラッチ回路 $RDL_1(E_{n1})$ の出力信号は、“L” (アンチフューズの絶縁体が未破壊状態) となっている。

【0066】

従って、選択信号 SEL_1 が “H” になると、アンド回路 AD_1 の出力信号は、“H” となる。また、プログラムスイッチ PSW は、 N チャンネル MOS トランジスタであるため、アンド回路 AD_1 の出力信号が “H” になると、オンになる。つまり、サブフューズセット SFS_1 を構成するフューズ素子 $FUSE$ の一端が、プログラムデータラッチ回路 $PDL(E_n)$, $PDL(D_{in1})$, \dots $PDL(D_{inm})$ に電氣的に接続される。

【0067】

ここで、プログラム時、例えば、プログラムデータが“1”のときは、NチャネルMOSトランジスタN1、PSWが、共に、オンとなるため、フューズ素子FUSEの両端には、高電圧Vfuseが印加される。その結果、例えば、フューズ素子（アンチフューズ）FUSEを構成する絶縁体が破壊され、データ“1”がプログラミングされる。

【0068】

また、例えば、プログラムデータが“0”のときは、NチャネルMOSトランジスタPSWは、オンであるが、NチャネルMOSトランジスタN1は、オフとなるため、フューズ素子FUSEの両端には、高電圧Vfuseが印加されない。その結果、例えば、フューズ素子（アンチフューズ）FUSEを構成する絶縁体が破壊されず、データ“0”がプログラミングされる。

【0069】**3. リードデータラッチ回路**

リードデータラッチ回路RDL1(En1), RDL1(Dout1), … RDL1(Doutm)は、(m+1)個のフューズ素子FUSEに対応して、(m+1)個だけ設けられている。リードデータラッチ回路RDL1(En1), RDL1(Dout1), … RDL1(Doutm)の回路構成は、例えば、図19に示すようになる。

【0070】

リードデータラッチ回路RDL1(En1), RDL1(Dout1), … RDL1(Doutm)の回路例については、既に、図19の説明において詳細に述べたため、ここでは、その説明については、省略する。

【0071】

重要な点は、例えば、パワーオン時に、リードデータラッチ回路RDL1(En1), RDL1(Dout1), … RDL1(Doutm)は、“0”状態、即ち、出力信号として、“L”を出力する状態に初期化されること、その直後、リードデータラッチ回路RDL1(En1), RDL1(Dout1), … RDL1(Doutm)の状態は、フューズデータに応じて変化することにある。

【0072】**4. リード制御回路**

リード制御回路RCNT1は、アンド回路AD2とリードスイッチRSWとから構成される。リード制御回路RCNT1は、フューズデータDin1, … Dinmがプログラムされるm個のフューズ素子FUSEに対応して、m個だけ設けられている。つまり、本例では、イネーブルビットEn1がプログラムされるフューズ素子FUSEに対応するリード制御回路は、存在しない。

【0073】

但し、イネーブルビットEn1がプログラムされるフューズ素子FUSEに対応するリード制御回路を設けても構わない。

【0074】

アンド回路AD2には、リセット信号RSTの反転信号、イネーブルビットデータEn1及びイネーブルビットデータEn2の反転信号が、それぞれ入力される。リセット信号RSTは、例えば、パワーオン時に、一時的に“H”となり、その後、“L”を維持し続ける。

【0075】

イネーブルビットデータEn1は、サブフューズセットSFS1に対するプログラミング前においては、“L”であり、サブフューズセットSFS1に対するプログラミング後においては、“H”である。イネーブルビットデータEn2は、サブフューズセットSFS2に対するプログラミング前においては、“L”であり、サブフューズセットSFS2に対するプログラミング後においては、“H”である。

【0076】

つまり、サブフューズセットSFS1にデータがプログラムされ、サブフューズセット

SFS2にデータがプログラムされていないときは、サブフューズセットSFS1にプログラムされているデータが最新のものとなり、例えば、パワーオン時に、リードスイッチRSWは、オンとなる。

【0077】

また、サブフューズセットSFS1, SFS2のいずれにも、データがプログラムされていないときは、リードスイッチRSWは、オンとなり得ない。サブフューズセットSFS1, SFS2の双方にデータがプログラムされているときは、サブフューズセットSFS2にプログラムされているデータが最新のものとなるため、リードスイッチRSWは、オンとなり得ない。

【0078】

[2] ブロックBLKi ($i = 2, 3, \dots, n-1$)

図4は、ブロックBLKiの回路例である。

1. プログラムデータラッチ回路

プログラムデータラッチ回路PDL(E_n), PDL(D_{in1}), \dots PDL(D_{inm})の構成については、既に、ブロックBLK1の項目において説明したため、ここでは、その説明を省略する。

【0079】

2. プログラム制御回路

プログラム制御回路PCNT1は、アンド回路AD1とプログラムスイッチPSWとから構成される。

【0080】

アンド回路AD1には、フューズセットの選択信号SEL1、イネーブルビットデータ $E_n(i-1)$ 及びイネーブルビットデータ E_{ni} の反転信号が、それぞれ入力される。フューズセットの選択信号SEL1は、既に説明したように、フューズセット1に対するプログラミングを実行するときに、“H”となる。

【0081】

イネーブルビットデータ $E_n(i-1)$ は、サブフューズセットSFS($i-1$)に対するプログラミング前においては、“L”であり、サブフューズセットSFS($i-1$)に対するプログラミング後においては、“H”である。また、イネーブルビットデータ E_{ni} は、サブフューズセットSFS i に対するプログラミング前においては、“L”であり、サブフューズセットSFS i に対するプログラミング後においては、“H”である。

【0082】

つまり、アンド回路AD1の出力信号は、サブフューズセットSFS($i-1$)に対するプログラミング後であって、サブフューズセットSFS i に対するプログラミング前において、“H”となり得る。

【0083】

アンド回路AD1の出力信号が“H”になると、プログラムスイッチPSWとしてのNチャネルMOSトランジスタは、オンになる。つまり、サブフューズセットSFS i を構成するフューズ素子FUSEの一端が、プログラムデータラッチ回路PDL(E_n), PDL(D_{in1}), \dots PDL(D_{inm})に電氣的に接続される。

【0084】

3. リードデータラッチ回路

リードデータラッチ回路RDL $i(E_{ni})$, RDL $i(D_{out1})$, \dots RDL $i(D_{outm})$ は、 $(m+1)$ 個のフューズ素子FUSEに対応して、 $(m+1)$ 個だけ設けられている。リードデータラッチ回路RDL $i(E_{ni})$, RDL $i(D_{out1})$, \dots RDL $i(D_{outm})$ の回路構成については、既に説明したため、ここでは、その説明については、省略する。

【0085】

4. リード制御回路

リード制御回路RCNT i は、アンド回路AD2とリードスイッチRSWとから構成さ

れる。リード制御回路 $RCNT\ i$ は、フューズデータ $Din\ 1, \dots, Din\ m$ がプログラムされる m 個のフューズ素子 $FUSE$ に対応して、 m 個だけ設けられている。

【0086】

アンド回路 $AD\ 2$ には、リセット信号 RST の反転信号、イネーブルビットデータ $En\ i$ 及びイネーブルビットデータ $En\ (i+1)$ の反転信号が、それぞれ入力される。リセット信号 RST は、例えば、パワーオン時に、一時的に“H”となり、その後、“L”を維持し続ける。

【0087】

イネーブルビットデータ $En\ i$ は、サブフューズセット $SFS\ i$ に対するプログラミング前においては、“L”であり、サブフューズセット $SFS\ i$ に対するプログラミング後においては、“H”である。イネーブルビットデータ $En\ (i+1)$ は、サブフューズセット $SFS\ (i+1)$ に対するプログラミング前においては、“L”であり、サブフューズセット $SFS\ (i+1)$ に対するプログラミング後においては、“H”である。

【0088】

つまり、サブフューズセット $SFS\ i$ にデータがプログラムされ、サブフューズセット $SFS\ (i+1)$ にデータがプログラムされていないときは、サブフューズセット $SFS\ i$ にプログラムされているデータが最新のものとなり、例えば、パワーオン時に、リードスイッチ RSW は、オンとなる。

【0089】

また、サブフューズセット $SFS\ i, SFS\ (i+1)$ のいずれにも、データがプログラムされていないときは、リードスイッチ RSW は、オンとなり得ない。サブフューズセット $SFS\ i, SFS\ (i+1)$ の双方にデータがプログラムされているときは、サブフューズセット $SFS\ (i+1)$ にプログラムされているデータが最新のものとなるため、リードスイッチ RSW は、オンとなり得ない。

【0090】

[3] ブロック $BLK\ n$

図5は、ブロック $BLK\ n$ の回路例である。

1. プログラムデータラッチ回路

プログラムデータラッチ回路 $PDL\ (En), PDL\ (Din\ 1), \dots, PDL\ (Din\ m)$ の構成については、既に、ブロック $BLK\ 1$ の項目において説明したため、ここでは、その説明を省略する。

【0091】

2. プログラム制御回路

プログラム制御回路 $PCNT\ 1$ は、アンド回路 $AD\ 1$ とプログラムスイッチ PSW とから構成される。

【0092】

アンド回路 $AD\ 1$ には、フューズセットの選択信号 $SEL\ 1$ 、イネーブルビットデータ $En\ (n-1)$ 及びイネーブルビットデータ $En\ n$ の反転信号が、それぞれ入力される。フューズセットの選択信号 $SEL\ 1$ は、既に説明したように、フューズセット 1 に対するプログラミングを実行するときに、“H”となる。

【0093】

イネーブルビットデータ $En\ (n-1)$ は、サブフューズセット $SFS\ (n-1)$ に対するプログラミング前においては、“L”であり、サブフューズセット $SFS\ (n-1)$ に対するプログラミング後においては、“H”である。また、イネーブルビットデータ $En\ n$ は、サブフューズセット $SFS\ n$ に対するプログラミング前においては、“L”であり、サブフューズセット $SFS\ n$ に対するプログラミング後においては、“H”である。

【0094】

つまり、アンド回路 $AD\ 1$ の出力信号は、サブフューズセット $SFS\ (n-1)$ に対するプログラミング後であって、サブフューズセット $SFS\ n$ に対するプログラミング前において、“H”となり得る。

【0095】

アンド回路AD1の出力信号が“H”になると、プログラムスイッチPSWとしてのNチャンネルMOSトランジスタは、オンになる。つまり、サブフューズセットSFS_nを構成するフューズ素子FUSEの一端が、プログラムデータラッチ回路PDL(E_n), PDL(D_{in1}), … PDL(D_{inm})に電氣的に接続される。

【0096】

3. リードデータラッチ回路

リードデータラッチ回路RDL_n(E_{nn}), RDL_n(D_{out1}), … RDL_n(D_{outm})は、(m+1)個のフューズ素子FUSEに対応して、(m+1)個だけ設けられている。リードデータラッチ回路RDL_n(E_{ni}), RDL_n(D_{out1}), … RDL_n(D_{outm})の回路構成については、既に説明したため、ここでは、その説明については、省略する。

【0097】

4. リード制御回路

リード制御回路RCNT_nは、アンド回路AD2とリードスイッチRSWとから構成される。リード制御回路RCNT_nは、フューズデータD_{in1}, … D_{inm}がプログラムされるm個のフューズ素子FUSEに対応して、m個だけ設けられている。

【0098】

アンド回路AD2には、リセット信号RSTの反転信号、イネーブルビットデータE_n及びイネーブルビットデータE_n(n+1)の反転信号が、それぞれ入力される。リセット信号RSTは、例えば、パワーオン時に、一時的に“H”となり、その後、“L”を維持し続ける。

【0099】

イネーブルビットデータE_{nn}は、サブフューズセットSFS_nに対するプログラミング前においては、“L”であり、サブフューズセットSFS_nに対するプログラミング後においては、“H”である。イネーブルビットデータE_n(n+1)は、常に、“L”に固定されている。

【0100】

つまり、サブフューズセットSFS_nにデータがプログラムされているときは、このサブフューズセットSFS_nにプログラムされているデータが最新のものとなるため、例えば、パワーオン時に、リードスイッチRSWは、オンとなる。

【0101】

なお、サブフューズセットSFS_nにデータがプログラムされていないときは、リードスイッチRSWは、オンとなり得ない。

【0102】

(3) 動作

次に、図1乃至図5におけるフューズ回路の動作について説明する。

【0103】

まず、フューズ回路の一般的な動作を説明する。

図6は、フューズ回路の動作の概要を示している。

パワーオンにより、例えば、LSIチップに電源が供給されると、まず、フューズセットの状態が検査される。即ち、フューズセットからフューズデータを読み出すフューズデータリードが実行される。この時、例えば、フューズセット内の一部のフューズ素子に記憶されたイネーブルビットデータも読み出される(ステップST1～ST2)。

【0104】

この後、例えば、LSIチップ内の制御回路又はLSIチップ外のCPUなどからフューズプログラムのためのコマンドが供給されると、フューズプログラムが実行される(ステップST3)。

【0105】

ここで、従来では、フューズプログラムは、1回のみであるか、又は、特殊な制御によ

りフューズセットを選択し、フューズプログラムを行う必要があるが、本発明の例では、フューズプログラムに関しては、選択動作などの特殊な制御なしに、フューズプログラムのためのコマンドを与えるだけで、自動的に、フューズデータの書き換えを行うことができる。

【0106】

以下、具体的に、本発明の例に関わるフューズ回路の動作について説明する。

なお、図1乃至図5に示すように、1つのフューズセットは、 n (n は、複数) 個のブロック、即ち、 n 個のサブフューズセットから構成され、各々のサブフューズセットは、1ビットのイネーブルビットデータを記憶する1個のフューズ素子と、 m (m は、複数) ビットのフューズデータを記憶する m 個のフューズ素子とから構成されるものとする。

【0107】

本発明の例では、フューズデータのリード／プログラム動作の制御及びフューズデータの書き換え動作については、フューズ回路内において、イネーブルビットデータに基づいて自動的に行われる。

【0108】

[初期状態]

まず、初期状態について説明する。

【表 1】

初期状態			SFS1	SFS2	...	SFSn		
	イネーブル ビット データ	En0	En1	En2	...	Enn	En(n+1)	
		1 "H"固定	0 "L"	0 "L"		0 "L"	0 "L"固定	
	PSW RSW			オン オフ	...	オフ オフ		
				オフ オフ				

【0109】
表 1 に示されるように、初期状態、即ち、1 つのフューズセットに対して、未だ、1 回もフューズプログラム動作を行っていない状態では、全てのサブフューズセット SFS1，SFS2，・・・SFSn 内のフューズ素子（例えば、アンチフューズ素子）は、非破壊状態にある。

【0110】

このため、例えば、パワーオンによりLSIチップに電源が供給されると、その1つのフューズセット内のサブフューズセットSFS1, SFS2, … SFSnから、それぞれ、イネーブルビットデータEn1, En2, … Ennとして、“0” (“L”レベル) が読み出され、これらがリードデータラッチ回路にラッチされる。

【0111】

ここで、最初のサブフューズセットSFS1を除く、残りの(n-1)個のサブフューズセットSFS2, … SFSnでは、図4及び図5に示すアンド回路AD1, AD2の出力信号は、全て、“L”となる。このため、プログラムスイッチPSW及びリードスイッチRSWは、いずれもオフ状態であり、これらサブフューズセットSFS2, … SFSnは、フューズプログラム又はフューズデータリードの対象とはならない。

【0112】

一方、最初(1番目)のサブフューズセットSFS1では、それより前のサブフューズセットが存在しないため、図3に示すアンド回路AD1に対する入力信号の一つとして、イネーブルビット信号En0が入力される。このイネーブルビット信号En0は、“H”に固定されているため、サブフューズセットSFS1では、アンド回路AD1の出力信号が“H”になる。このため、プログラムスイッチPSWは、オン状態であり、サブフューズセットSFS1は、プログラムデータラッチ回路に電氣的に接続される。

【0113】

なお、この段階では、図3に示すアンドAD2の出力信号は、“L”であるため、リードスイッチRSWは、オフ状態となっている。

【0114】

従って、パワーオン後、例えば、フューズデータをプログラムするためのコマンドがLSIチップに供給され、かつ、イネーブルビットデータEn (“1”) 及びプログラムデータ (“0” 又は “1”) がLSIチップに供給されると、これらデータは、プログラムデータラッチ回路にラッチされる。

【0115】

また、プログラムデータラッチ回路にラッチされたイネーブルビットデータEn及びプログラムデータの値に基づいて、サブフューズセットSFS1内のフューズ素子FUSEに対するプログラミングが実行される。

【0116】

つまり、イネーブルビットデータEnは、“1” (= “H”) であるため、イネーブルビットデータEnを記憶するフューズ素子FUSEの両端には、高電圧Vfuseが印加され、その結果、フューズ素子FUSEは、破壊される。また、プログラムデータに関しては、“0” 又は “1” であるため、“0” のときは、フューズ素子FUSEは、未破壊のままとなり、“1” のときは、フューズ素子FUSEは、破壊される。

【0117】

サブフューズセットSFS1に対するフューズプログラミングが終了すると、サブフューズセットSFS1内のイネーブルビットデータEn1の値は、“0” (= “L”) から “1” (= “H”) に変化する。

【0118】

従って、サブフューズセットSFS1では、図3に示すアンド回路AD1の出力信号が “H” から “L” に変化し、プログラムスイッチPSWは、オフ状態になる。これとほぼ同時に、サブフューズセットSFS1では、図3に示すアンドAD2の出力信号が “L” から “H” に変化し、リードスイッチRSWは、オン状態になる。

【0119】

[フューズプログラム(1回目)終了後の初期状態]。

【表 2】

フューズプログラム(1回目)終了後の初期状態					
イネーブル ビット データ	<div><div>En0</div><div>1</div><div>"H"固定</div></div>	SFS1	En1	1	"H"
		SFS2	En2	0	"L"
	
PSW RSW	<div><div>Enn</div><div>0</div><div>"L"</div></div>	SFSn	Enn	0	"L"
	
		<div><div>En(n+1)</div><div>0</div><div>"L"固定</div></div>			

【0 1 2 0】
表 2 に示されるように、フューズプログラム（1 回目）終了後の初期状態では、最初のサブフューズセット S F S 1 内のフューズ素子（イネーブルビットデータ）は、破壊状態

にあり、残りのサブフューズセット SFS_2, \dots, SFS_n 内のフューズ素子（イネーブルビットデータ）は、非破壊状態にある。

【0121】

このため、例えば、再度、パワーオンにより LSI チップに電源が供給されると、サブフューズセット SFS_1 からは、イネーブルビットデータ En_1 として、“1”（“H”レベル）が読み出され、また、サブフューズセット SFS_2, \dots, SFS_n からは、イネーブルビットデータ En_2, \dots, En_n として、“0”（“L”レベル）が読み出される。そして、これらデータは、リードデータラッチ回路にラッチされる。

【0122】

ここで、最初（1番目）のサブフューズセット SFS_1 では、イネーブルビットデータ En_0, En_1 が“H”であるため、図3に示すアンド回路 AD_1 の出力信号は、全て、“L”となる。このため、プログラムスイッチ PSW は、オフ状態となる。一方、イネーブルビットデータ En_1 が“H”、イネーブルビットデータ En_2 が“L”であるため、図3に示すアンド AD_2 の出力信号は、“H”となる。このため、リードスイッチ RSW は、オン状態となる。

【0123】

従って、最初のサブフューズセット SFS_1 に関しては、既に、フューズプログラムされているため、フューズプログラムの対象にはならないが、最新のフューズデータを記憶しているため、フューズデータリードの対象となる。

【0124】

2番目のサブフューズセット SFS_2 では、イネーブルビットデータ En_1 が“H”、イネーブルビットデータ En_2 が“L”であるため、図4に示すアンド AD_1 の出力信号は、全て、“H”となる。このため、プログラムスイッチ PSW は、オン状態となる。一方、イネーブルビットデータ En_2, En_3 が“L”であるため、図4に示すアンド AD_2 の出力信号は、“L”となる。このため、リードスイッチ RSW は、オフ状態となる。

【0125】

従って、2番目のサブフューズセット SFS_2 に関しては、フューズプログラムの対象（データ書き換えの対象）になるが、フューズデータリードの対象にはならない。

【0126】

3番目以降のサブフューズセット SFS_3, \dots, SFS_n では、イネーブルビットデータ En_2, En_3, \dots, En_n が全て“L”であるため、図4及び図5に示すアンド回路 AD_1, AD_2 の出力信号は、全て、“L”となる。このため、プログラムスイッチ PSW 及びリードスイッチ RSW は、いずれもオフ状態であり、これらサブフューズセット SFS_3, \dots, SFS_n は、フューズプログラム又はフューズデータリードの対象とはならない。

【0127】

従って、パワーオン後、まず、サブフューズセット SFS_1 のフューズデータが読み出され、フューズセットの状態が確認される。この時、イネーブルビットデータ En_0, En_1 が“H”、イネーブルビットデータ En_2, \dots, En_n が“L”に確定し、サブフューズセット SFS_2 がプログラムデータラッチ回路に電氣的に接続される。

【0128】

この後、例えば、フューズデータをプログラムするためのコマンドが LSI チップに供給され、かつ、イネーブルビットデータ En （“1”）及びプログラムデータ（“0”又は“1”）が LSI チップに供給されると、これらデータは、プログラムデータラッチ回路にラッチされる。

【0129】

また、プログラムデータラッチ回路にラッチされたイネーブルビットデータ En 及びプログラムデータの値に基づいて、サブフューズセット SFS_2 内のフューズ素子 $FUSE$ に対するプログラミングが実行される。

【0130】

つまり、イネーブルビットデータ E_n は、“1” (= “H”) であるため、イネーブルビットデータ E_n を記憶するフューズ素子 $FUSE$ の両端には、高電圧 V_{fuse} が印加され、その結果、フューズ素子 $FUSE$ は、破壊される。また、プログラムデータに関しては、“0” 又は “1” であるため、“0” のときは、フューズ素子 $FUSE$ は、未破壊のままとなり、“1” のときは、フューズ素子 $FUSE$ は、破壊される。

【0131】

サブフューズセット $SFS2$ に対するフューズプログラミングが終了すると、サブフューズセット $SFS2$ 内のイネーブルビットデータ E_{n2} の値は、“0” (= “L”) から “1” (= “H”) に変化する。

【0132】

従って、サブフューズセット $SFS2$ では、図 4 に示すアンド回路 $AD1$ の出力信号が “H” から “L” に変化し、プログラムスイッチ PSW は、オフ状態になる。これとほぼ同時に、サブフューズセット $SFS2$ では、図 4 に示すアンド $AD2$ の出力信号が “L” から “H” に変化し、リードスイッチ RSW は、オン状態になる。

【0133】

[フューズプログラム (2 回目) 終了後の初期状態]。

【表 3】

フューズプログラム(2回目)終了後の初期状態

	SFS1	SFS2	SFS3	...	SFSn	
	En1	En2	En3	...	Enn	En(n+1)
イネーブル ビット データ	1 "H"	1 "H"	0 "L"	...	0 "L"	0 "L"固定
PSW	オフ	オフ	オン	...	オフ	
RSW	オフ	オン	オフ	...	オフ	

【0134】

表3に示されるように、フューズプログラム（2回目）終了後の初期状態では、サブフューズセットSFS1、SFS2内のフューズ素子（イネーブルビットデータ）は、破壊

状態にあり、残りのサブフューズセット SFS_3, \dots, SFS_n 内のフューズ素子（イネーブルビットデータ）は、非破壊状態にある。

【0135】

このため、例えば、再度、パワーオンにより LSI チップに電源が供給されると、サブフューズセット SFS_1, SFS_2 からは、イネーブルビットデータ E_{n1}, E_{n2} として、“1”（“H”レベル）が読み出され、また、サブフューズセット SFS_3, \dots, SFS_n からは、イネーブルビットデータ E_{n3}, \dots, E_{nn} として、“0”（“L”レベル）が読み出される。そして、これらデータは、リードデータラッチ回路にラッチされる。

【0136】

この場合の動作は、「フューズプログラム（2回目）終了後の初期状態」の欄で説明したのと同じとなるため、詳細な説明は、省略する。ただ、ここでは、サブフューズセット SFS_2 に最新のプログラムデータが記憶されているため、サブフューズセット SFS_2 のリードスイッチ RSW がオン状態となる。また、データ書き換えにより、新たにプログラムデータをフューズ素子に記憶させるときは、サブフューズセット SFS_3 がプログラムの対象となる。従って、サブフューズセット SFS_3 内のプログラムスイッチ PSW がオンとなる。

【0137】

[フューズプログラム（n回目）終了後の初期状態]。

【表 4】

フューズプログラム(n回目)終了後の初期状態						
		SFS1	SFS2	SFS3	...	SFSn
イネーブル ビット データ	En0	En1	En2	En3	...	Enn
	1 "H"固定	1 "H"	1 "H"	1 "H"		1 "H"
						En(n+1) 0 "L"固定
PSW RSW		オフ	オフ	オフ	...	オフ
		オフ	オフ	オフ		オン

【 0 1 3 8 】

表4に示されるように、フューズプログラム（ n 回目）終了後の初期状態では、全てのサブフューズセット $SFS1, SFS2, \dots, SFSn$ 内のフューズ素子（イネーブル

ビットデータ)は、破壊状態にある。

【0139】

このため、例えば、パワーオンによりLSIチップに電源が供給されると、サブフューズセットSFS1, SFS2, … SFSnからは、イネーブルビットデータEn1, En2, … Ennとして、“1”(“H”レベル)が読み出される。そして、これらデータは、リードデータラッチ回路にラッチされる。

【0140】

ここで、サブフューズセットSFSnの次のサブフューズセットは存在しないため、本発明の例に関わるフューズ回路のデータ書き換え回数は、最大でn回となる。また、サブフューズセットSFSnの次のサブフューズセットは存在しないため、図5に示すように、イネーブルビットデータEn(n+1)として、“L”(固定)の反転信号、即ち、“H”を、サブフューズセットSFSn内のアンド回路AD2に与える。従って、サブフューズセットSFSn内のリードスイッチRSWがオン状態となる。

【0141】

(4) まとめ

このように、本発明の例に関わるフューズ回路では、1つのフューズセットをn(nは、複数)個のサブフューズセットから構成することで、n回のフューズデータの書き換えを可能にする。

【0142】

また、このような構成にしても、フューズデータのリード/プログラム又は書き換えは、フューズデータをフューズ回路に入力するだけで、特殊な制御なく、自動的かつ容易に行うことができる。

【0143】

つまり、プログラムに関しては、サブフューズセット内のイネーブルビットデータの値をフューズ回路内でフィードバックさせることにより、どのサブフューズセットにデータを書き込んだらよいかを自動的に決めることができる。また、リードに関しても、サブフューズセット内のイネーブルビットデータの値をフューズ回路内でフィードバックさせることにより、どのサブフューズセットからデータを読み出したらよいかを自動的に決めることができる。

【0144】

以上をまとめると、本発明の例に関わるフューズ回路では、フューズセットに対するプログラムは、最大でn回行うことができ、かつ、データ書き換えを行う度に、プログラミングの対象は、サブフューズセットSFS1からサブフューズセットSFSnに向かって、1つずつ、順次、ずれていく。

【0145】

このような動作の制御は、フューズ回路内でのみ行われるため、例えば、このフューズ回路にプログラムデータを与える制御回路から見ると、フューズ回路には、プログラムデータを与えるだけでよく、実質的に、1回のみしかプログラムを行えない従来のフューズセットに対するプログラム手法をそのまま適用できる。

【0146】

つまり、本発明の例に関わるフューズ回路では、n個のサブフューズセットのうち、どのサブフューズセットに対してプログラミングを実行し、どのフューズセットに対してフューズデータのリードを行うかは、フューズ回路内で自動的に行われるため、データ書き換えのための特殊な制御は、不要である。

【0147】

なお、本例では、全てのサブフューズセットSFS1, SFS2, … SFSnのレイアウトを同じにするため、最初のサブフューズセットSFS1では、“H”に固定されたイネーブルビットデータEn0を使用し、かつ、最後のサブフューズセットSFSnでは、“L”に固定されたイネーブルビットデータEn(n+1)を使用している。

【0148】

しかし、最初のサブフューズセット $SFS1$ では、イネーブルビットデータ E_{n0} をなくし、アンド回路 $AD1$ に入力される信号を 2 つとし、また、最後のサブフューズセット SFS_n では、イネーブルビットデータ $E_n (n+1)$ をなくし、アンド回路 $AD2$ に入力される信号を 2 つとしてもよい。

【0149】

4. 第2実施例

第2実施例は、上述の第1実施例と比べると、プログラムに関する回路については同じであるが、フューズデータのリードに関する回路が大きく異なっている。そのポイントは、フューズデータに対するリードデータラッチ回路を、 n (n は、複数) 個のサブフューズセットで共有化した点にある。

【0150】

第1実施例では、サブフューズセットごとに、フューズデータに対するリードデータラッチ回路を設けているが、この場合、サブフューズセットの数、即ち、データ書き換え回数の最大値が大きくなればなる程、リードデータラッチ回路の数も増え、フューズ回路のレイアウト面積が大きくなる。

【0151】

そこで、フューズ回路のレイアウト面積の増大を抑えるため、第2実施例では、リードデータラッチ回路を、 n 個のサブフューズセットに対して、共通に、1 個だけ設ける。そして、どのサブフューズセットをリードデータラッチ回路に電氣的に接続するかは、第1実施例と同様に、フューズ回路内のリード制御回路により決定する。

【0152】

なお、第2実施例においても、共通化できるのは、内部回路で使用するデータに対するリードデータラッチ回路であり、サブフューズセットが有効か否かを示すイネーブルビットデータに対するリードデータラッチ回路は、サブフューズセットごとに、個別に設ける必要がある。

【0153】

なぜなら、本発明の例に関わるフューズ回路では、イネーブルビットデータの値に基づいて、データの書き換え動作、即ち、どのサブフューズセットに対してプログラムを実行し、どのサブフューズセットからフューズデータを読み出すか、を制御しているからである。

【0154】

つまり、本発明の例に関わるフューズ回路では、パワーオン後、全てのサブフューズセット内のイネーブルビットデータの値を直ちに認識する必要があるため、イネーブルビットデータに対するリードデータラッチ回路に関しては、サブフューズセットごとに、個別に設けておく。

【0155】

なお、第1実施例では、パワーオン時におけるフューズデータの読み出し動作、即ち、リードデータラッチ回路へのラッチ動作が 1 回で済むのに対して、第2実施例では、パワーオン時におけるフューズデータの読み出し動作、即ち、リードデータラッチ回路へのラッチ動作が 2 回必要になる。

【0156】

具体的には、第1実施例では、パワーオン時に、全てのサブフューズセットのフューズデータを一度ラッチしておき（1 回のみの読み出し）、この後、イネーブルビットデータに基づいて、選択されたサブフューズセットのみからフューズデータを出力する、ということが可能である。

【0157】

しかし、第2実施例では、パワーオン後、まず、イネーブルビットデータをラッチし、サブフューズセットの選択を行う必要がある（1 回目の読み出し）。この後、選択されたサブフューズセットのフューズデータをラッチし、かつ、出力する（2 回目の読み出し）、という動作が必要である。

【0158】

以下、第2実施例に関わるフューズ回路について詳細に説明する。

【0159】

(1) フューズセット

図7は、本発明の第2実施例に関わるフューズ回路の主要部を示している。

【0160】

同図に示す構成は、図1の1つのフューズセットに対応している。1つのフューズセットは、 n (n は、複数) 個のフューズブロック $BLK1$, $BLK2$, \dots $BLKn$ を有する。また、各フューズブロック $BLKi$ ($i=1, 2, \dots, n$) は、 $(m+1)$ 個のフューズ素子 (m は、複数) から構成されるサブフューズセット $SFSi$ と、このサブフューズセット $SFSi$ に対応して設けられる $(m+1)$ 個のプログラム制御回路 $PCNTi$ 及びイネーブルビットデータをラッチするための1個のリードデータラッチ回路 $RDL(E ni)$ とを有する。

【0161】

また、各フューズブロック $BLKi$ は、リード制御回路 $RCNTi$ を有している。リード制御回路 $RCNTi$ は、イネーブルビットデータを除く、 m ビットのフューズデータを記憶する m 個のサブフューズセット $SFSi$ に対応して、 m 個だけ設けられる。

【0162】

さらに、本例では、1つのフューズセットは、 n 個のフューズブロック $BLK1$, $BLK2$, \dots $BLKn$ に共有されるリードデータラッチ回路 RDL を有している。リードデータラッチ回路 RDL は、リード制御回路 $RCNTi$ により選択された1個のフューズブロック $BLKi$ に対して、イネーブルビットデータを除く、 m ビットのフューズデータをラッチする。

【0163】

プログラムデータラッチ回路 PDL は、イネーブルビットデータ En 及びプログラムデータ $Din1$, $Din2$, \dots $Dinm$ を一時的にラッチする。これら入力データ En , $Din1$, $Din2$, \dots $Dinm$ は、選択された1個のフューズブロック $BLKi$ に転送される。

【0164】

ここで、 n 個のフューズブロック $BLK1$, $BLK2$, \dots $BLKn$ は、直列接続される。そして、プログラム制御回路 $PCNT1$, $PCNT2$, \dots $PCNTn$ は、イネーブルビットデータに基づいて、 n 個のフューズブロック $BLK1$, $BLK2$, \dots $BLKn$ のうちから、プログラミングの対象となる1個のフューズブロック $BLKj$ を選択する。

【0165】

即ち、 i ($i=1, 2, \dots, n$) 番目のフューズブロック $BLKi$ では、プログラム制御回路 $PCNTi$ は、フューズブロック $BLKi$ 内のイネーブルビットデータ Eni の値と、1つ前の $(i-1)$ 番目のフューズブロック $BLK(i-1)$ 内のイネーブルビットデータ $En(i-1)$ の値とに基づいて、データプログラムに関し、フューズブロック $BLKi$ の選択／非選択を決定する。

【0166】

但し、最初(1番目)のフューズブロック $BLK1$ では、1つ前のフューズブロックが存在しない。また、最初にプログラムの対象となるフューズブロックは、フューズブロック $BLK1$ である。そこで、最初のフューズブロック $BLK1$ では、プログラム制御回路 $PCNT1$ は、フューズブロック $BLK1$ 内のイネーブルビットデータ $En1$ の値と、入力データ $En0$ の値 (“H” 固定) とに基づいて、フューズブロック $BLK1$ の選択／非選択を決定する。

【0167】

なお、各フューズブロック $BLKi$ 内のイネーブルビットデータ Eni は、例えば、パワーオンの後、直ちに、リードラッチ回路 $RDL(E ni)$ にラッチされ、イネーブルビ

ットデータ E_{ni} として、各フューズブロック BLK_i にフィードバックされる。

【0168】

また、リード制御回路 $RCNT_1$, $RCNT_2$, \dots , $RCNT_n$ は、イネーブルビットデータに基づいて、 n 個のフューズブロック BLK_1 , BLK_2 , \dots , BLK_n のうちから、データリードの対象となる 1 個のフューズブロック BLK_j を選択する。

【0169】

即ち、 i ($i = 1, 2, \dots, n$) 番目のフューズブロック BLK_i では、リード制御回路 $RCNT_i$ は、フューズブロック BLK_i 内のイネーブルビットデータ E_{ni} の値と、1 つ後の ($i + 1$) 番目のフューズブロック $BLK_{(i+1)}$ 内のイネーブルビットデータ $E_{n(i+1)}$ の値とに基づいて、データリードに関し、フューズブロック BLK_i の選択／非選択を決定する。

【0170】

但し、最後 (n 番目) のフューズブロック BLK_n では、1 つ後のフューズブロックが存在しない。また、フューズブロック BLK_n は、最後にプログラムの対象となるフューズブロックである。そこで、最後のフューズブロック BLK_n では、リード制御回路 $RCNT_n$ は、フューズブロック BLK_n 内のイネーブルビットデータ E_{nn} の値と、入力データ $E_{n(n+1)}$ の値 (“L” 固定) とに基づいて、フューズブロック BLK_n の選択／非選択を決定する。

【0171】

選択された 1 個のフューズブロック BLK_i からは、フューズデータが出力される。このフューズデータは、全てのブロック BLK_1 , BLK_2 , \dots , BLK_n で共有化されたリードデータラッチ回路 RDL に転送され、そこにラッチされる。

【0172】

(2) 回路例

図 8 乃至図 11 は、図 7 のフューズセットを構成するブロック BLK_i の回路例を示している。

【0173】

[1] ブロック BLK_1

図 8 は、ブロック BLK_1 の回路例である。

1. プログラムデータラッチ回路

プログラムデータラッチ回路 $PDL(E_n)$ は、イネーブルビットデータ E_n をラッチする。プログラムデータラッチ回路 $PDL(D_{in1})$, \dots , $PDL(D_{inm})$ は、 m ビットのプログラムデータ D_{in1} , \dots , D_{inm} に対応して m 個だけ設けられ、かつ、プログラムデータ D_{in1} , \dots , D_{inm} をラッチする。

【0174】

プログラムデータラッチ回路 $PDL(E_n)$, $PDL(D_{in1})$, \dots , $PDL(D_{inm})$ は、全て、同じ回路構成を有している。

【0175】

インバータ I_1 及びクロックドインバータ CI_1 により、ラッチ回路が構成される。ラッチ回路の出力信号は、ノア回路 NR_1 の一方の入力端子に入力される。ノア回路 NR_1 の他方の入力端子には、プログラム信号 $PROG$ の反転信号 $bPROG$ が入力される。

【0176】

プログラム時には、プログラム信号 $PROG$ が “H” となり、かつ、その反転信号が “L” となる。このため、プログラム時には、ノア回路 NR_1 は、ラッチ回路にラッチされたデータに応じた出力信号を出力する。

【0177】

例えば、ラッチ回路にラッチされたデータが “1”、即ち、クロックドインバータ CI_1 の出力信号が “L” のときは、ノア回路 NR_1 の出力信号は、“H” となる。その結果、 N チャンネル MOS トランジスタ N_1 は、オンになる。また、ラッチ回路にラッチされたデータが “0”、即ち、クロックドインバータ CI_1 の出力信号が “H” のときは、ノア

回路NR1の出力信号は、“L”となる。その結果、NチャネルMOSトランジスタN1は、オフになる。

【0178】

2. プログラム制御回路

プログラム制御回路PCNT1は、アンド回路AD1とプログラムスイッチPSWとから構成される。

【0179】

アンド回路AD1には、フューズセットの選択信号SEL1、イネーブルビットデータEn0 (“H”固定) 及びイネーブルビットデータEn1の反転信号が、それぞれ入力される。フューズセットの選択信号SEL1は、例えば、図1及び図7に示すように、フューズセット1に対するプログラミングを実行するときに、“H”となる。イネーブルビットデータEn1は、サブフューズセットSFS1に対するプログラミング前においては、“L”、即ち、リードデータラッチ回路RDL1 (En1) の出力信号は、“L” (アンチフューズの絶縁体が未破壊状態) となっている。

【0180】

従って、選択信号SEL1が“H”になると、アンド回路AD1の出力信号は、“H”となる。また、プログラムスイッチPSWは、NチャネルMOSトランジスタであるため、アンド回路AD1の出力信号が“H”になると、オンになる。つまり、サブフューズセットSFS1を構成するフューズ素子FUSEの一端が、プログラムデータラッチ回路PDL (En), PDL (Din1), … PDL (Dinm) に電氣的に接続される。

【0181】

ここで、プログラム時、例えば、プログラムデータが“1”のときは、NチャネルMOSトランジスタN1, PSWが、共に、オンとなるため、フューズ素子FUSEの両端には、高電圧Vfuseが印加される。その結果、例えば、フューズ素子 (アンチフューズ) FUSEを構成する絶縁体が破壊され、データ“1”がプログラミングされる。

【0182】

また、例えば、プログラムデータが“0”のときは、NチャネルMOSトランジスタPSWは、オンであるが、NチャネルMOSトランジスタN1は、オフとなるため、フューズ素子FUSEの両端には、高電圧Vfuseが印加されない。その結果、例えば、フューズ素子 (アンチフューズ) FUSEを構成する絶縁体が破壊されず、データ“0”がプログラミングされる。

【0183】

3. リードデータラッチ回路

イネーブルビットデータEn1をラッチするためのリードデータラッチ回路RDL (En1) は、フューズ素子FUSEに対応して、1個だけ設けられる。また、イネーブルビットデータEn1を除く、残りのmビットのフューズデータをラッチするためのリードデータラッチ回路RDL (Dout1), RDL (Dout2), … RDL (Doutm) は、m個のフューズ素子FUSEに対応して、m個だけ設けられる。

【0184】

但し、リードデータラッチ回路RDL (Dout1), RDL (Dout2), … RDL (Doutm) は、上述したように、1つのフューズセット内の複数のブロックに対して、共通に、1個だけ設けられる。

【0185】

リードデータラッチ回路RDL (En1), RDL (Dout1), … RDL (Doutm) の回路構成については、例えば、図19に示すようになる。リードデータラッチ回路RDL (En1), RDL (Dout1), … RDL (Doutm) の回路構成については、既に、図19の説明において詳細に述べたため、ここでは、その説明については、省略する。

【0186】

4. リード制御回路

リード制御回路RCNT1は、アンド回路AD2とリードスイッチRSWとから構成される。リード制御回路RCNT1は、フューズデータDin1, … Dinmがプログラムされるm個のフューズ素子FUSEに対応して、m個だけ設けられている。つまり、本例では、イネーブルビットEn1がプログラムされるフューズ素子FUSEに対応するリード制御回路は、存在しない。

【0187】

但し、イネーブルビットEn1がプログラムされるフューズ素子FUSEに対応するリード制御回路を設けても構わない。

【0188】

アンド回路AD2には、リセット信号RSTの反転信号、イネーブルビットデータEn1及びイネーブルビットデータEn2の反転信号が、それぞれ入力される。リセット信号RSTは、例えば、パワーオン時に、一時的に“H”となり、その後、“L”を維持し続ける。

【0189】

イネーブルビットデータEn1は、サブフューズセットSFS1に対するプログラミング前においては、“L”であり、サブフューズセットSFS1に対するプログラミング後においては、“H”である。イネーブルビットデータEn2は、サブフューズセットSFS2に対するプログラミング前においては、“L”であり、サブフューズセットSFS2に対するプログラミング後においては、“H”である。

【0190】

つまり、サブフューズセットSFS1にデータがプログラムされ、サブフューズセットSFS2にデータがプログラムされていないときは、サブフューズセットSFS1にプログラムされているデータが最新のものとなり、例えば、パワーオン時に、リードスイッチRSWは、オンとなる。

【0191】

また、サブフューズセットSFS1, SFS2のいずれにも、データがプログラムされていないときは、リードスイッチRSWは、オンとなり得ない。サブフューズセットSFS1, SFS2の双方にデータがプログラムされているときは、サブフューズセットSFS2にプログラムされているデータが最新のものとなるため、リードスイッチRSWは、オンとなり得ない。

【0192】

[2] ブロックBLKi (i = 2, 3, … n-1)

図9は、ブロックBLKiの回路例である。

1. プログラムデータラッチ回路

プログラムデータラッチ回路PDL(En), PDL(Din1), … PDL(Dinm)の構成については、既に、ブロックBLK1の項目において説明したため、ここでは、その説明を省略する。

【0193】

2. プログラム制御回路

プログラム制御回路PCNT1は、アンド回路AD1とプログラムスイッチPSWとから構成される。

【0194】

アンド回路AD1には、フューズセットの選択信号SEL1、イネーブルビットデータEn(i-1)及びイネーブルビットデータEniの反転信号が、それぞれ入力される。フューズセットの選択信号SEL1は、既に説明したように、フューズセット1に対するプログラミングを実行するときに、“H”となる。

【0195】

イネーブルビットデータEn(i-1)は、サブフューズセットSFS(i-1)に対するプログラミング前においては、“L”であり、サブフューズセットSFS(i-1)に対するプログラミング後においては、“H”である。また、イネーブルビットデータE

$n i$ は、サブフューズセット $S F S i$ に対するプログラミング前においては、“L”であり、サブフューズセット $S F S i$ に対するプログラミング後においては、“H”である。

【0196】

つまり、アンド回路 $A D 1$ の出力信号は、サブフューズセット $S F S (i-1)$ に対するプログラミング後であって、サブフューズセット $S F S i$ に対するプログラミング前において、“H”となり得る。

【0197】

アンド回路 $A D 1$ の出力信号が“H”になると、プログラムスイッチ $P S W$ としての N チャンネル $M O S$ トランジスタは、オンになる。つまり、サブフューズセット $S F S i$ を構成するフューズ素子 $F U S E$ の一端が、プログラムデータラッチ回路 $P D L (E n)$ 、 $P D L (D i n 1)$ 、 \dots $P D L (D i n m)$ に電氣的に接続される。

【0198】

3. リードデータラッチ回路

リードデータラッチ回路 $R D L (E n i)$ 、 $R D L (D o u t 1)$ 、 \dots $R D L (D o u t m)$ は、 $(m+1)$ 個のフューズ素子 $F U S E$ に対応して、 $(m+1)$ 個だけ設けられている。但し、上述したように、イネーブルビットデータ $E n i$ を除く、他の m ビットのフューズデータをラッチするための m 個のリードデータラッチ回路 $R D L (D o u t 1)$ 、 $R D L (D o u t 2)$ 、 \dots $R D L (D o u t m)$ については、1つのフューズセット内の複数のブロックに共有化される。

【0199】

リードデータラッチ回路 $R D L (E n i)$ 、 $R D L (D o u t 1)$ 、 \dots $R D L (D o u t m)$ の回路構成については、既に説明したため、ここでは、その説明については、省略する。

【0200】

4. リード制御回路

リード制御回路 $R C N T i$ は、アンド回路 $A D 2$ とリードスイッチ $R S W$ とから構成される。リード制御回路 $R C N T i$ は、フューズデータ $D i n 1$ 、 \dots $D i n m$ がプログラムされる m 個のフューズ素子 $F U S E$ に対応して、 m 個だけ設けられている。

【0201】

アンド回路 $A D 2$ には、リセット信号 $R S T$ の反転信号、イネーブルビットデータ $E n i$ 及びイネーブルビットデータ $E n (i+1)$ の反転信号が、それぞれ入力される。リセット信号 $R S T$ は、例えば、パワーオン時に、一時的に“H”となり、その後、“L”を維持し続ける。

【0202】

イネーブルビットデータ $E n i$ は、サブフューズセット $S F S i$ に対するプログラミング前においては、“L”であり、サブフューズセット $S F S i$ に対するプログラミング後においては、“H”である。イネーブルビットデータ $E n (i+1)$ は、サブフューズセット $S F S (i+1)$ に対するプログラミング前においては、“L”であり、サブフューズセット $S F S (i+1)$ に対するプログラミング後においては、“H”である。

【0203】

つまり、サブフューズセット $S F S i$ にデータがプログラムされ、サブフューズセット $S F S (i+1)$ にデータがプログラムされていないときは、サブフューズセット $S F S i$ にプログラムされているデータが最新のものとなり、例えば、パワーオン時に、リードスイッチ $R S W$ は、オンとなる。

【0204】

また、サブフューズセット $S F S i$ 、 $S F S (i+1)$ のいずれにも、データがプログラムされていないときは、リードスイッチ $R S W$ は、オンとなり得ない。サブフューズセット $S F S i$ 、 $S F S (i+1)$ の双方にデータがプログラムされているときは、サブフューズセット $S F S (i+1)$ にプログラムされているデータが最新のものとなるため、リードスイッチ $R S W$ は、オンとなり得ない。

【0205】

[3] ブロックBLKn

図10は、ブロックBLKnの回路例である。

1. プログラムデータラッチ回路

プログラムデータラッチ回路PDL(En), PDL(Din1), … PDL(Dinm)の構成については、既に、ブロックBLK1の項目において説明したため、ここでは、その説明を省略する。

【0206】

2. プログラム制御回路

プログラム制御回路PCNT1は、アンド回路AD1とプログラムスイッチPSWとから構成される。

【0207】

アンド回路AD1には、フューズセットの選択信号SEL1、イネーブルビットデータEn(n-1)及びイネーブルビットデータEnnの反転信号が、それぞれ入力される。フューズセットの選択信号SEL1は、既に説明したように、フューズセット1に対するプログラミングを実行するときに、“H”となる。

【0208】

イネーブルビットデータEn(n-1)は、サブフューズセットSFS(n-1)に対するプログラミング前においては、“L”であり、サブフューズセットSFS(n-1)に対するプログラミング後においては、“H”である。また、イネーブルビットデータEnnは、サブフューズセットSFSnに対するプログラミング前においては、“L”であり、サブフューズセットSFSnに対するプログラミング後においては、“H”である。

【0209】

つまり、アンド回路AD1の出力信号は、サブフューズセットSFS(n-1)に対するプログラミング後であって、サブフューズセットSFSnに対するプログラミング前において、“H”となり得る。

【0210】

アンド回路AD1の出力信号が“H”になると、プログラムスイッチPSWとしてのNチャンネルMOSトランジスタは、オンになる。つまり、サブフューズセットSFSnを構成するフューズ素子FUSEの一端が、プログラムデータラッチ回路PDL(En), PDL(Din1), … PDL(Dinm)に電氣的に接続される。

【0211】

3. リードデータラッチ回路

リードデータラッチ回路RDL(Enn), RDL(Dout1), … RDL(Doutm)は、(m+1)個のフューズ素子FUSEに対応して、(m+1)個だけ設けられている。但し、上述したように、イネーブルビットデータEniを除く、他のmビットのフューズデータをラッチするためのm個のリードデータラッチ回路RDL(Dout1), RDL(Dout2), … RDL(Doutm)については、1つのフューズセット内の複数のブロックに共有化される。

【0212】

リードデータラッチ回路RDL(Eni), RDL(Dout1), … RDL(Doutm)の回路構成については、既に説明したため、ここでは、その説明については、省略する。

【0213】

4. リード制御回路

リード制御回路RCNTnは、アンド回路AD2とリードスイッチRSWとから構成される。リード制御回路RCNTnは、フューズデータDin1, … Dinmがプログラムされるm個のフューズ素子FUSEに対応して、m個だけ設けられている。

【0214】

アンド回路AD2には、リセット信号RSTの反転信号、イネーブルビットデータEn

n 及びイネーブルビットデータ $E_n (n+1)$ の反転信号が、それぞれ入力される。リセット信号 RST は、例えば、パワーオン時に、一時的に“H”となり、その後、“L”を維持し続ける。

【0215】

イネーブルビットデータ E_{nn} は、サブフューズセット SFS_n に対するプログラミング前においては、“L”であり、サブフューズセット SFS_n に対するプログラミング後においては、“H”である。イネーブルビットデータ $E_n (n+1)$ は、常に、“L”に固定されている。

【0216】

つまり、サブフューズセット SFS_n にデータがプログラムされているときは、このサブフューズセット SFS_n にプログラムされているデータが最新のものとなるため、例えば、パワーオン時に、リードスイッチ RSW は、オンとなる。

【0217】

なお、サブフューズセット SFS_n にデータがプログラムされていないときは、リードスイッチ RSW は、オンとなり得ない。

【0218】

(3) 動作

次に、図7乃至図10におけるフューズ回路の動作について説明する。

第2実施例に関わるフューズ回路の動作は、全体としては、第1実施例に関わるフューズ回路の動作とほぼ同じであるが、パワーオン時におけるフューズデータのラッチ動作が2回になる点が異なる。

【0219】

以下、具体的に、本発明の例に関わるフューズ回路の動作について説明する。

なお、図7乃至図10に示すように、1つのフューズセットは、 n (n は、複数) 個のブロック、即ち、 n 個のサブフューズセットから構成され、各々のサブフューズセットは、1ビットのイネーブルビットデータを記憶する1個のフューズ素子と、 m (m は、複数) ビットのフューズデータを記憶する m 個のフューズ素子とから構成されるものとする。

【0220】

[初期状態]

初期状態、即ち、1つのフューズセットに対して、未だ、1回もフューズプログラム動作を行っていない状態では、全てのサブフューズセット $SFS_1, SFS_2, \dots, SFS_n$ 内のフューズ素子 (例えば、アンチフューズ素子) は、非破壊状態にある。

【0221】

このため、例えば、パワーオンにより LSI チップに電源が供給されると、その1つのフューズセット内のサブフューズセット $SFS_1, SFS_2, \dots, SFS_n$ から、それぞれ、イネーブルビットデータ $E_{n1}, E_{n2}, \dots, E_{nn}$ として、“0” (“L”レベル) が読み出され、これらがリードデータラッチ回路 $RDL(E_{n1}), RDL(E_{n2}), \dots, RDL(E_{nn})$ にラッチされる (1回目のラッチ動作)。

【0222】

ここで、最初のサブフューズセット SFS_1 を除く、残りの $(n-1)$ 個のサブフューズセット SFS_2, \dots, SFS_n では、図9及び図10に示すアンド回路 AD_1, AD_2 の出力信号は、全て、“L”となる。このため、プログラムスイッチ PSW 及びリードスイッチ RSW は、いずれもオフ状態であり、これらサブフューズセット SFS_2, \dots, SFS_n は、フューズプログラム又はフューズデータリードの対象とはならない。

【0223】

一方、最初 (1番目) のサブフューズセット SFS_1 では、それより前のサブフューズセットが存在しないため、図8に示すアンド回路 AD_1 に対する入力信号の一つとして、イネーブルビット信号 E_{n0} が入力される。このイネーブルビット信号 E_{n0} は、“H”に固定されているため、サブフューズセット SFS_1 では、アンド回路 AD_1 の出力信号が“H”になる。このため、プログラムスイッチ PSW は、オン状態であり、サブフュー

ズセットSFS1は、プログラムデータラッチ回路に電氣的に接続される。

【0224】

なお、この段階では、図8に示すアンドAD2の出力信号は、“L”であるため、リードスイッチRSWは、オフ状態となっている。

【0225】

イネーブルビットデータE_{n1}, E_{n2}, … E_{nn}がリードデータラッチ回路RDL(E_{n1}), RDL(E_{n2}), … RDL(E_{nn})にラッチされると、各ブロック内のリード制御回路RCNT1, 2, … nは、その値に基づいて、リードスイッチRSWのオン/オフを決定する。

【0226】

ここでは、いずれのブロック内のサブフューズセットSFS1, SFS2, … SFS_nにも、フューズデータが書き込まれていないため、全てのブロックにおいて、リードスイッチRSWは、オフ状態である。このため、この後、フューズデータのラッチ動作(2回目のラッチ動作)が実行されるが、ラッチデータ自体が存在しないため、全てのリードデータラッチ回路RDL(Dout1), RDL(Dout2), … RDL(Dout_m)のデータは、“0”となる。

【0227】

パワーオン後、例えば、フューズデータをプログラムするためのコマンドがLSIチップに供給され、かつ、イネーブルビットデータE_n(“1”)及びプログラムデータ(“0”又は“1”)がLSIチップに供給されると、これらデータは、プログラムデータラッチ回路にラッチされる。

【0228】

また、プログラムデータラッチ回路にラッチされたイネーブルビットデータE_n及びプログラムデータの値に基づいて、サブフューズセットSFS1内のフューズ素子FUSEに対するプログラミングが実行される。

【0229】

つまり、イネーブルビットデータE_nは、“1”(=“H”)であるため、イネーブルビットデータE_nを記憶するフューズ素子FUSEの両端には、高電圧V_{fuse}が印加され、その結果、フューズ素子FUSEは、破壊される。また、プログラムデータに関しては、“0”又は“1”であるため、“0”のときは、フューズ素子FUSEは、未破壊のままとなり、“1”のときは、フューズ素子FUSEは、破壊される。

【0230】

サブフューズセットSFS1に対するフューズプログラミングが終了すると、サブフューズセットSFS1内のイネーブルビットデータE_{n1}の値は、“0”(=“L”)から“1”(=“H”)に変化する。

【0231】

従って、サブフューズセットSFS1では、図8に示すアンド回路AD1の出力信号が“H”から“L”に変化し、プログラムスイッチPSWは、オフ状態になる。これとほぼ同時に、サブフューズセットSFS1では、図8に示すアンドAD2の出力信号が“L”から“H”に変化し、リードスイッチRSWは、オン状態になる。

【0232】

[フューズプログラム(1回目)終了後の初期状態]

フューズプログラム(1回目)終了後の初期状態では、最初のサブフューズセットSFS1内のフューズ素子(イネーブルビットデータ)は、破壊状態にあり、残りのサブフューズセットSFS2, … SFS_n内のフューズ素子(イネーブルビットデータ)は、非破壊状態にある。

【0233】

このため、例えば、再度、パワーオンによりLSIチップに電源が供給されると、サブフューズセットSFS1からは、イネーブルビットデータE_{n1}として、“1”(“H”レベル)が読み出され、また、サブフューズセットSFS2, … SFS_nからは、イ

ネーブルビットデータ E_{n2} , \dots , E_{nn} として、“0” (“L” レベル) が読み出される。そして、これらデータは、リードデータラッチ回路 $RDL(E_{n1})$, $RDL(E_{n2})$, \dots , $RDL(E_{nn})$ にラッチされる (1 回目のラッチ動作)。

【0234】

ここで、最初 (1 番目) のサブフューズセット $SFS1$ では、イネーブルビットデータ E_{n0} , E_{n1} が “H” であるため、図 8 に示すアンド回路 $AD1$ の出力信号は、全て、“L” となる。このため、プログラムスイッチ PSW は、オフ状態となる。一方、イネーブルビットデータ E_{n1} が “H”、イネーブルビットデータ E_{n2} が “L” であるため、図 8 に示すアンド $AD2$ の出力信号は、“H” となる。このため、リードスイッチ RSW は、オン状態となる。

【0235】

従って、最初のサブフューズセット $SFS1$ に関しては、既に、フューズプログラムされているため、フューズプログラムの対象にはならないが、最新のフューズデータを記憶しているため、フューズデータリードの対象となる。

【0236】

2 番目のサブフューズセット $SFS2$ では、イネーブルビットデータ E_{n1} が “H”、イネーブルビットデータ E_{n2} が “L” であるため、図 9 に示すアンド $AD1$ の出力信号は、全て、“H” となる。このため、プログラムスイッチ PSW は、オン状態となる。一方、イネーブルビットデータ E_{n2} , E_{n3} が “L” であるため、図 9 に示すアンド $AD2$ の出力信号は、“L” となる。このため、リードスイッチ RSW は、オフ状態となる。

【0237】

従って、2 番目のサブフューズセット $SFS2$ に関しては、フューズプログラムの対象 (データ書き換えの対象) になるが、フューズデータリードの対象にはならない。

【0238】

3 番目以降のサブフューズセット $SFS3$, \dots , $SFSn$ では、イネーブルビットデータ E_{n2} , E_{n3} , \dots , E_{nn} が全て “L” であるため、図 9 及び図 10 に示すアンド回路 $AD1$, $AD2$ の出力信号は、全て、“L” となる。このため、プログラムスイッチ PSW 及びリードスイッチ RSW は、いずれもオフ状態であり、これらサブフューズセット $SFS3$, \dots , $SFSn$ は、フューズプログラム又はフューズデータリードの対象とはならない。

【0239】

このように、イネーブルビットデータ E_{n1} , E_{n2} , \dots , E_{nn} がリードデータラッチ回路 $RDL(E_{n1})$, $RDL(E_{n2})$, \dots , $RDL(E_{nn})$ にラッチされた後、各ブロック内のリード制御回路 $RCNT1$, 2 , \dots , n は、その値に基づいて、リードスイッチ RSW のオン/オフを決定する。

【0240】

ここでは、サブフューズセット $SFS1$ にフューズデータが書き込まれているため、最初のブロック $BLK1$ においては、リードスイッチ RSW は、オン状態である。このため、この後、フューズデータのラッチ動作 (2 回目のラッチ動作) が実行されるが、このラッチ動作では、リード制御回路 $RCNT1$, $RCNT2$, \dots , $RCNTn$ により最初のブロック $BLK1$ が選択され、サブフューズセット $SFS1$ に記憶されたラッチデータが、リードデータラッチ回路 $RDL(Dout1)$, $RDL(Dout2)$, \dots , $RDL(Doutm)$ にラッチされる。

【0241】

また、パワーオン後、イネーブルビットデータ E_{n0} , E_{n1} が “H”、イネーブルビットデータ E_{n2} , \dots , E_{nn} が “L” となっているため、サブフューズセット $SFS2$ がプログラムデータラッチ回路に電氣的に接続される。

【0242】

従って、例えば、フューズデータをプログラムするためのコマンドが LSI チップに供給され、かつ、イネーブルビットデータ E_n (“1”) 及びプログラムデータ (“0”) 又

は“1”)がLSIチップに供給されると、これらデータは、プログラムデータラッチ回路にラッチされる。

【0243】

また、プログラムデータラッチ回路にラッチされたイネーブルビットデータ E_n 及びプログラムデータの値に基づいて、サブフューズセットSFS2内のフューズ素子FUSEに対するプログラミングが実行される。

【0244】

サブフューズセットSFS2に対するフューズプログラミングが終了すると、サブフューズセットSFS2内のイネーブルビットデータ E_{n2} の値は、“0”(=“L”)から“1”(=“H”)に変化する。

【0245】

従って、サブフューズセットSFS2では、図9に示すアンド回路AD1の出力信号が“H”から“L”に変化し、プログラムスイッチPSWは、オフ状態になる。これとほぼ同時に、サブフューズセットSFS2では、図9に示すアンドAD2の出力信号が“L”から“H”に変化し、リードスイッチRSWは、オン状態になる。

【0246】

[フューズプログラム(n回目)終了後の初期状態]

フューズプログラム(n回目)終了後の初期状態では、全てのサブフューズセットSFS1, SFS2, … SFSn内のフューズ素子(イネーブルビットデータ)は、破壊状態にある。

【0247】

このため、例えば、パワーオンによりLSIチップに電源が供給されると、サブフューズセットSFS1, SFS2, … SFSnからは、イネーブルビットデータ E_{n1} , E_{n2} , … E_{nn} として、“1”(“H”レベル)が読み出される。そして、これらデータは、リードデータラッチ回路にラッチされる(1回目のラッチ動作)。

【0248】

ここで、サブフューズセットSFSnの次のサブフューズセットは存在しないため、本発明の例に関わるフューズ回路のデータ書き換え回数は、最大でn回となる。また、サブフューズセットSFSnの次のサブフューズセットは存在しないため、図10に示すように、イネーブルビットデータ $E_{n(n+1)}$ として、“L”(固定)の反転信号、即ち、“H”を、サブフューズセットSFSn内のアンド回路AD2に与える。従って、サブフューズセットSFSn内のリードスイッチRSWがオン状態となる。

【0249】

ここでは、全てのサブフューズセットSFS1, SFS2, … SFSnにフューズデータが書き込まれているため、最後のブロックBLKnにおけるリードスイッチRSWのみがオン状態になる。

【0250】

このため、この後、フューズデータのラッチ動作(2回目のラッチ動作)が実行されるが、このラッチ動作では、リード制御回路RCNT1, RCNT2, … RCNTnにより最後のブロックBLKnが選択され、サブフューズセットSFSnに記憶されたラッチデータが、リードデータラッチ回路RDL(Dout1), RDL(Dout2), … RDL(Doutm)にラッチされる。

【0251】

(4) まとめ

このように、第2実施例に関わるフューズ回路においても、第1実施例に関わるフューズ回路と同様に、1つのフューズセットをn(nは、複数)個のサブフューズセットから構成することで、n回のフューズデータの書き換えが可能になる。

【0252】

また、このような構成にしても、フューズデータのリード/プログラム又は書き換えは、フューズデータをフューズ回路に入力するだけで、特殊な制御なく、自動的かつ容易に

行うことができる。

【0253】

5. 応用例

本発明の例に関わるフューズ回路を有するLSIチップ及びそのLSIを利用したシステムの例について説明する。

【0254】

図11は、システムLSIのチップレイアウトの一例を示している。

LSIチップ上には、CPU、ロジック回路、SRAM及びDRAMが形成される。また、LSIチップ上の任意の位置に、本回路、即ち、本発明の例に関わるフューズ回路が配置される。チップの外部から見た場合には、チップに与える信号や制御方法などに関して、従来と本発明の例とで、何ら変わるところはない。つまり、本発明の例では、複雑な制御なしに、フューズセットのデータ書き換えを可能にする。

【0255】

なお、LSIチップに関し、そのチップ内に組み込まれる機能（ブロック）としては、本例に限定されるものではなく、当然に、図11の例に対して、LSIチップ内に組み込まれる機能（ブロック）の追加、減少、変更などが可能である。

【0256】

図12は、汎用LSIのチップレイアウトの一例を示している。

本例では、汎用メモリのチップレイアウトを示す。メモリチップ上には、メモリセルアレイ及びその周辺回路が形成される。また、メモリチップ上の任意の位置に、本回路、即ち、本発明の例に関わるフューズ回路が配置される。チップの外部から見た場合には、チップに与える信号や制御方法などに関して、従来と本発明の例とで、何ら変わるところはない。つまり、本発明の例では、複雑な制御なしに、フューズセットに対するデータ書き換えを可能にする。

【0257】

なお、本例では、汎用メモリのレイアウトについて説明したが、例えば、その他のLSIチップ、例えば、メモリとロジック回路とを混載したメモリ混載ロジックLSIなどに、本発明の例に関わるフューズ回路を適用することもできる。

【0258】

図13は、接触式ICカードの概略を示している。

プラスチックカード10には、コネクタ部11及びICカード用MPU12が搭載されている。コネクタ部11は、外部電極を有しており、この外部電極を外部装置（リーダ／ライタ）18に直接接触させることによりデータなどのやりとりを行う。一般に、コネクタ部11とICカード用MPU12は、モジュール化され、ICモジュールとなっており、このICモジュールをプラスチックカード10のエンボス領域に嵌め込むことでICカードが出来上がる。

【0259】

ICカード用MPU12は、制御部13、演算部14、ROM15及びRAM16を備えている。ROM15には、データ処理用のプログラムが保存されている。例えば、このROM15に、本発明の例に関わるフューズ回路を適用できる。また、RAM16は、データの一時記憶用として用いられる。EEPROM17は、データ保存用として用いられる。

【0260】

図14は、無線式ICカードの概略を示している。

プラスチックカード20には、アンテナ21及びICカード用MPU22が内蔵されている。アンテナ21は、外部装置（リーダ／ライタ）30との間でデータなどのやりとりを行うためのものである。ICカード用MPU22は、変復調回路23、入出力制御回路24、CPU25、ROM26、RAM27及びバス29を備えている。

【0261】

アンテナ21において受信された無線信号は、変復調回路23を経由して入出力制御回

路 24 に入力される。また、ROM 26 には、データ処理用のプログラムが保存されている。例えば、この ROM 26 に、本発明の例に関わるフューズ回路を適用できる。SRAM 27 は、データの一時記憶用として用いられる。EEPROM 28 は、データ保存用として用いられる。

【0262】

なお、本発明の例に関わるフューズ回路は、1 枚のカードで様々なアプリケーションを処理できるマルチカード（例えば、コンビカードなど）内に搭載されるチップに適用することもできる。

【0263】

図 15 は、IC カードの外観を示している。

プラスチックカード 10 は、一定の厚さを有し、その表面の一部には、エンボス領域が設けられている。IC モジュール 53 には、IC（チップ）が搭載されている。IC モジュール 53 の IC が搭載される側の面に対して反対側の面には、外部端子（電極）54 が形成されている。IC モジュール 53 は、外部端子 54 が剥き出しになるようにしてエンボス領域に嵌め込まれる。

【0264】

図 16 は、IC モジュールを示している。また、図 17 は、図 16 の X V I I - X V I I 線に沿う断面図である。なお、図 16 において、外部端子は、省略している。

IC モジュール 53 の一面側には、IC（チップ）55 が搭載されている。IC 55 は、樹脂 56 により覆われている。また、IC モジュール 53 の他面側には、外部電極 54 が形成されている。

【0265】

6. まとめ

本発明の例に関わるフューズ回路によれば、標準 CMOS プロセスで形成でき、かつ、1 つのフューズセットに対して、フューズデータの書き換えを、特殊な制御なしに、自動的に行うことができる。

【0266】

ところで、本発明の例では、フューズデータの書き換えは、パッケージング（アセンブリ）工程後におけるユーザ側での書き換えを主眼に置いているが、当然に、その他の時期、例えば、パッケージング工程前に、フューズデータの書き換えをする必要性が生じたならば、そのときに、本発明を適用することもできる。

【0267】

本発明の例に関わるフューズ回路は、ロジック回路（ASIC など）とメモリ回路（DRAM, SRAM など）とを混載したメモリ混載ロジック LSI に有効である。例えば、このような混載 LSI において、本発明の例に関わるフューズ回路を、メモリ回路（リダンダンシ回路）における不良アドレス（リダンダンシデータ）のプログラムに適用することができる。

【0268】

本発明の例に関わるフューズ回路は、リダンダンシ回路における不良アドレスのリード／プログラムに限られず、様々なデータのプログラム、特に、容量の少ないデータ（例えば、512 キロビット以下）のプログラムに効果的である。

【0269】

例えば、本発明の例に関わるフューズ回路は、回路動作に関するトリミングデータ、チップ ID、セキュリティコード、さらには、携帯電話の液晶ディスプレイのコントラストに関するデータなどをプログラムする場合に適用できる。

【0270】

本発明の例に関わるフューズ回路では、各フューズブロック内のサブフューズセットのビット数は、全て等しく設定されているが、異なってもよい。また、1 つのフューズセット（図 1 参照）内の各フューズブロック内のサブフューズセットのビット数を等しく設定し、異なるフューズセット間で、プログラムできるビット数を異ならせるようにして

も構わない。

【0271】

7. その他

本発明は、上述の形態に限定されるものではなく、その要旨を逸脱しない範囲で、構成要素を変形して具体化できる。また、上述の形態に開示されている複数の構成要素の適宜な組み合わせにより種々の発明を構成できる。例えば、上述の形態に開示される全構成要素から幾つかの構成要素を削除してもよいし、異なる形態の構成要素を適宜組み合わせてもよい。

【産業上の利用可能性】

【0272】

本発明の例に関わるフューズ回路は、特に、少ないデータを不揮発に記憶する必要があると共に、少数回ではあるが、これらデータを書き換える必要もある混載LSI、システムLSI、ロジックLSI、メモリLSIなどの半導体集積回路に有効である。

【図面の簡単な説明】

【0273】

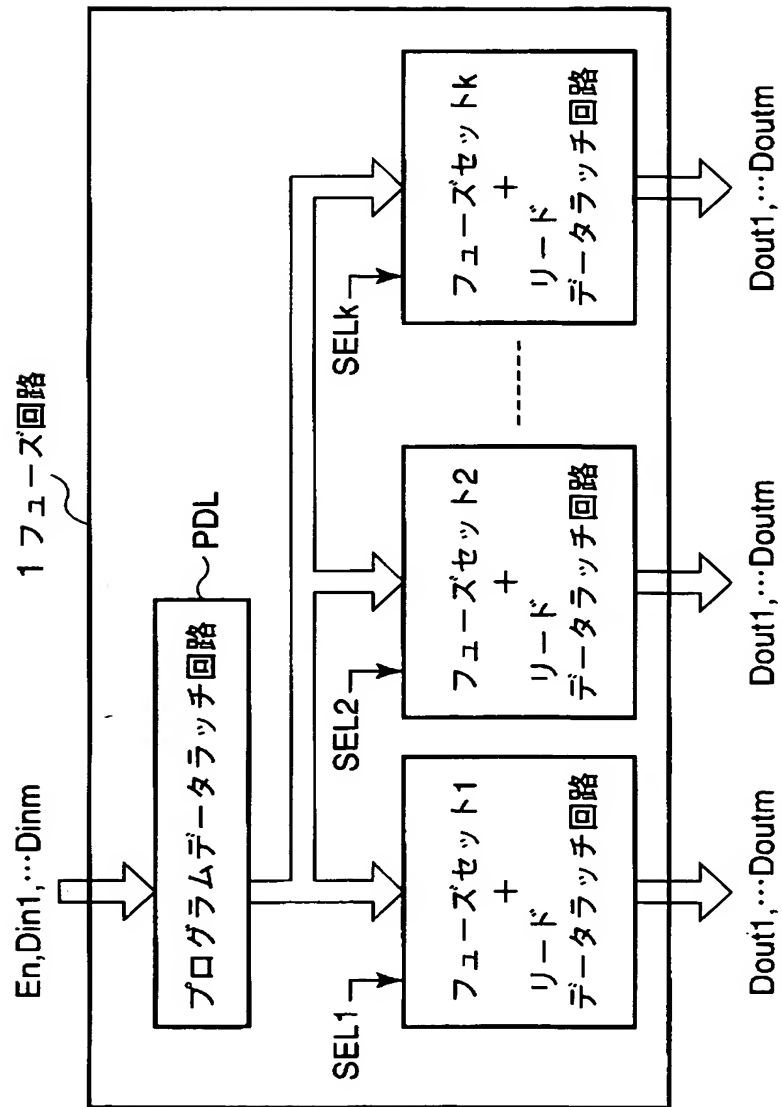
- 【図1】 フューズ回路の例を示す図。
- 【図2】 第1実施例に関わるフューズ回路の概要を示す図。
- 【図3】 第1実施例に関わるフューズ回路を示す回路図。
- 【図4】 第1実施例に関わるフューズ回路を示す回路図。
- 【図5】 第1実施例に関わるフューズ回路を示す回路図。
- 【図6】 パワーオン時のチップ動作を示す図。
- 【図7】 第2実施例に関わるフューズ回路の概要を示す図。
- 【図8】 第2実施例に関わるフューズ回路を示す回路図。
- 【図9】 第2実施例に関わるフューズ回路を示す回路図。
- 【図10】 第2実施例に関わるフューズ回路を示す回路図。
- 【図11】 システムLSIのチップレイアウトの例を示す図。
- 【図12】 汎用LSIのチップレイアウトの例を示す図。
- 【図13】 接触式ICカードの例を示す図。
- 【図14】 無線式ICカードの例を示す図。
- 【図15】 ICカードの外観を示す図。
- 【図16】 ICモジュールの例を示す図。
- 【図17】 図16のXVII-XVII線に沿う断面図。
- 【図18】 従来のフューズ回路の概要を示す図。
- 【図19】 従来のフューズ回路を示す回路図。

【符号の説明】

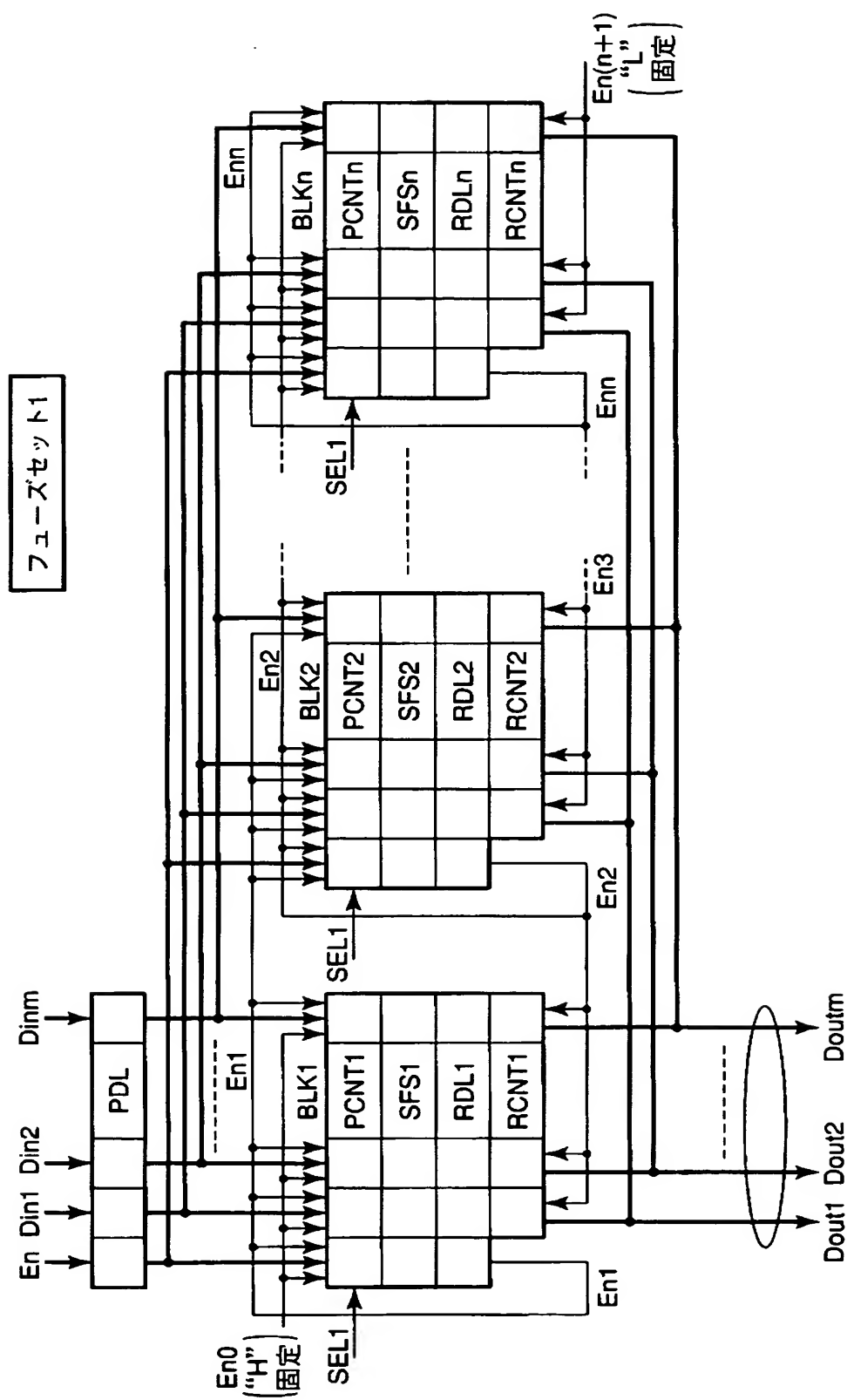
【0274】

1： フューズ回路、 PDL： プログラムデータラッチ回路、 PCNT1, PCNT2, … PCNTn： プログラム制御回路、 SFS1, SFS2, … SFSn： サブフューズセット、 RDL1, RDL2, … RDLn： リードデータラッチ回路、 RCNT1, RCNT2, … RCNTn： リード制御回路、 I1： インバータ、 CI1： クロックドインバータ、 NR1： ノア回路、 PSW： プログラムスイッチ、 RSW： リードスイッチ、 AD1, AD2： アンド回路、 FUSE： フューズ素子。

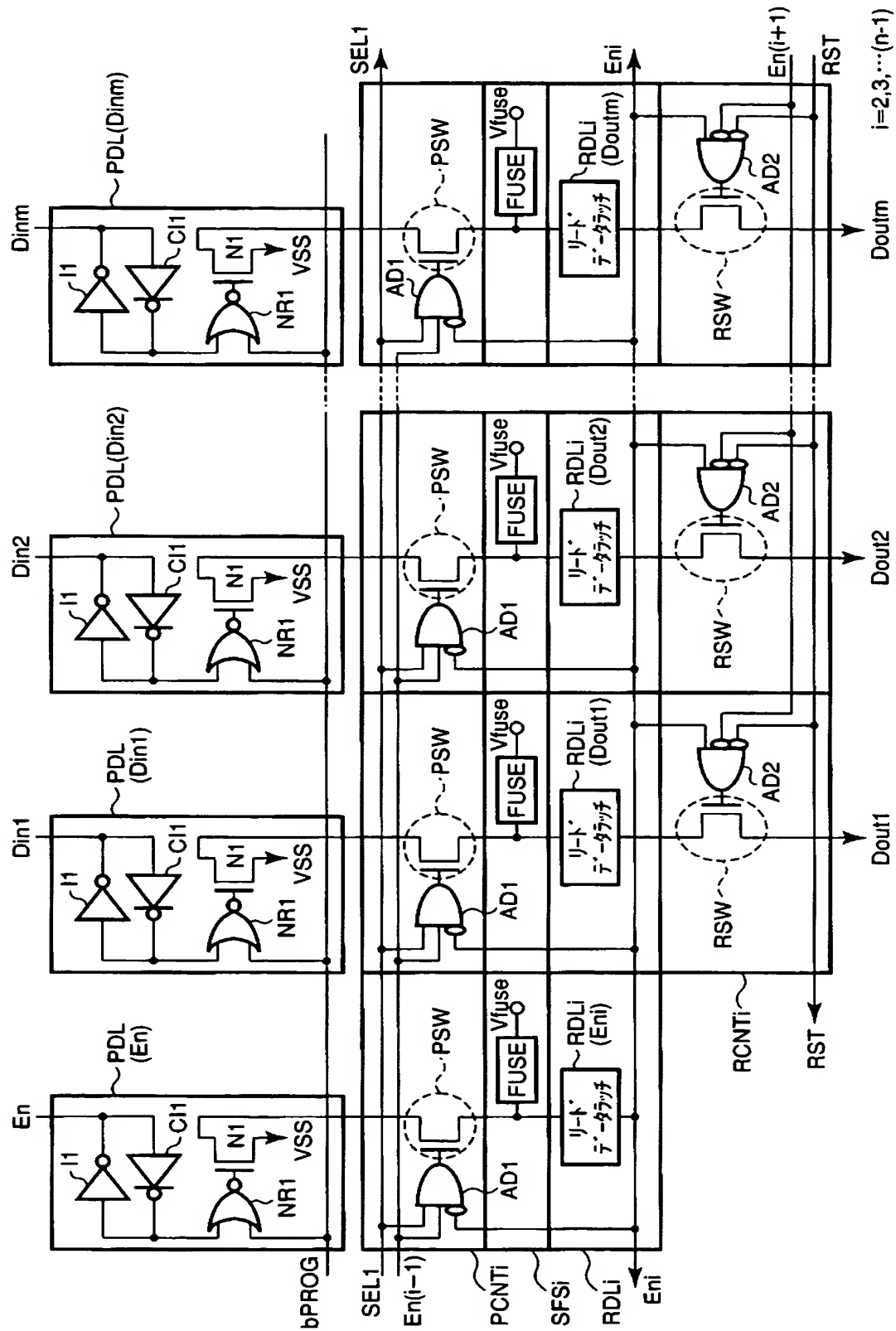
【書類名】 図面
【図 1】



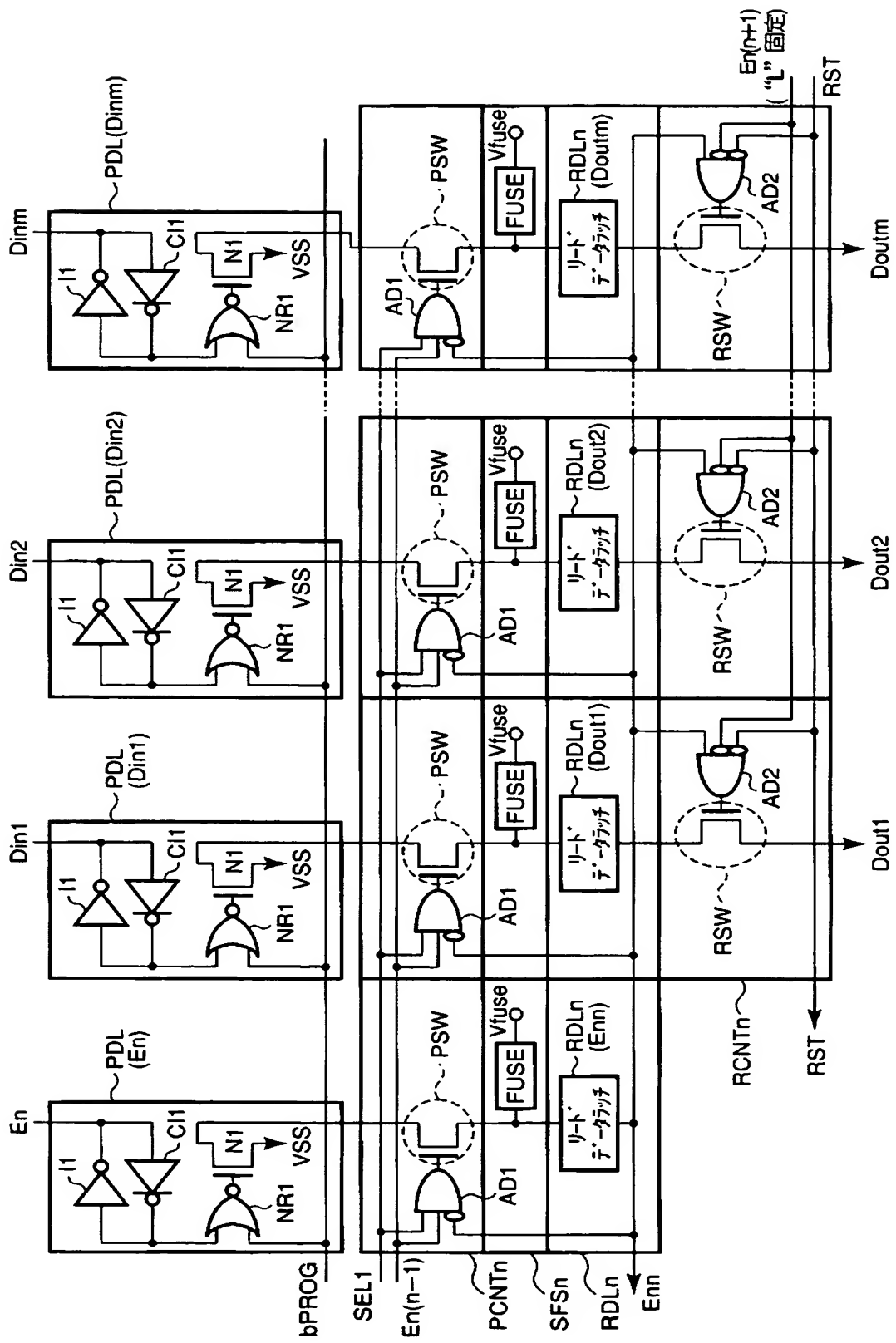
【図 2】



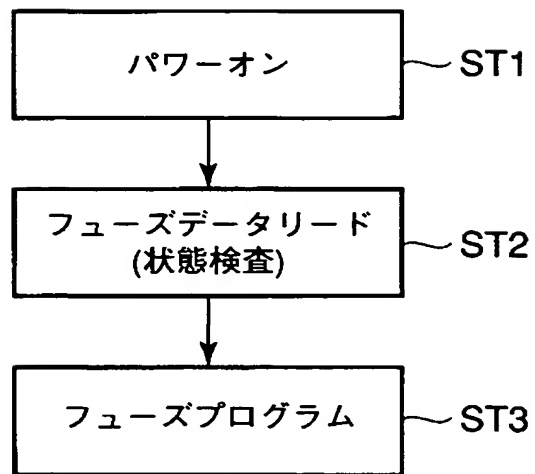
【図 4】



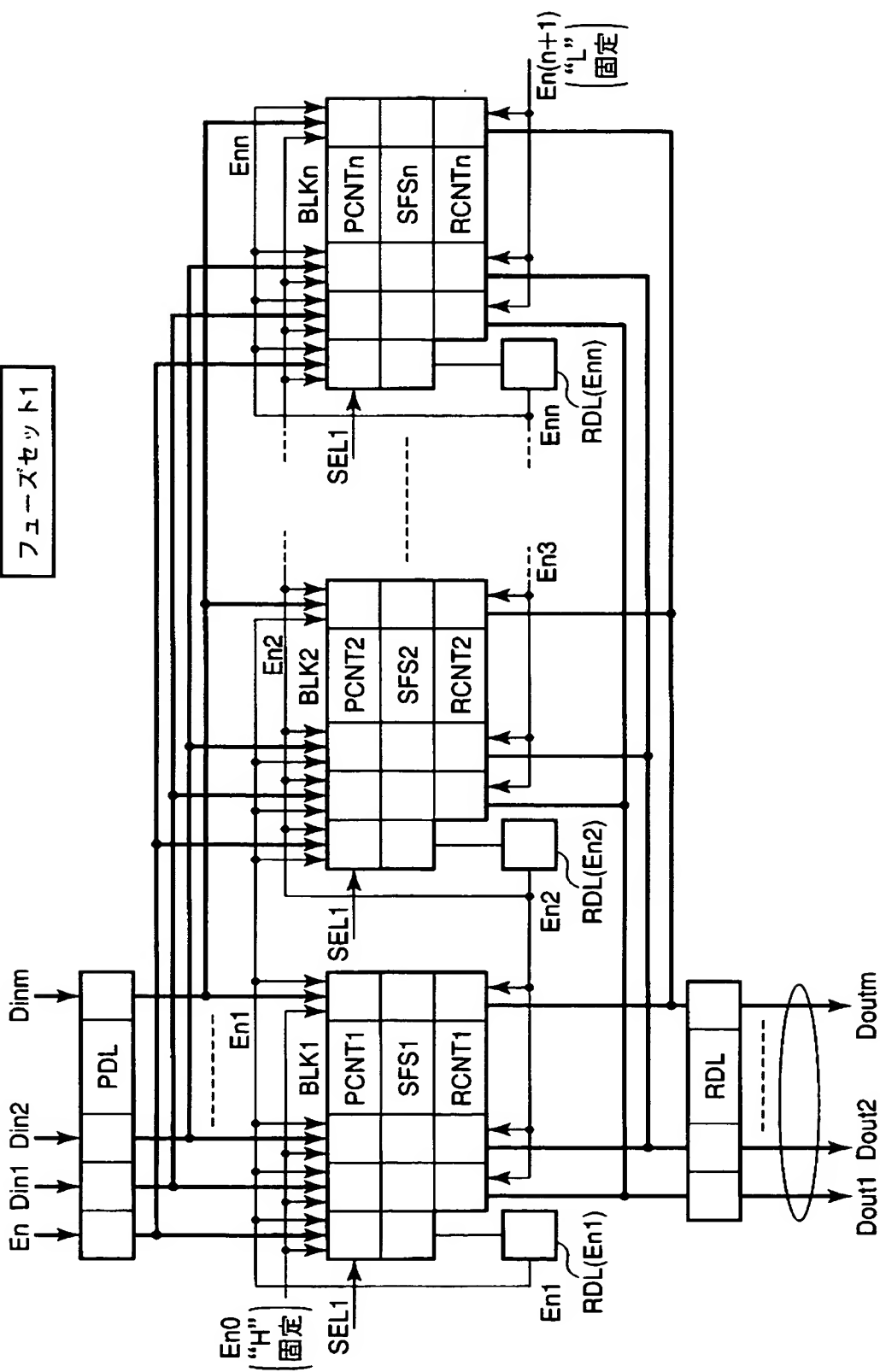
【図5】



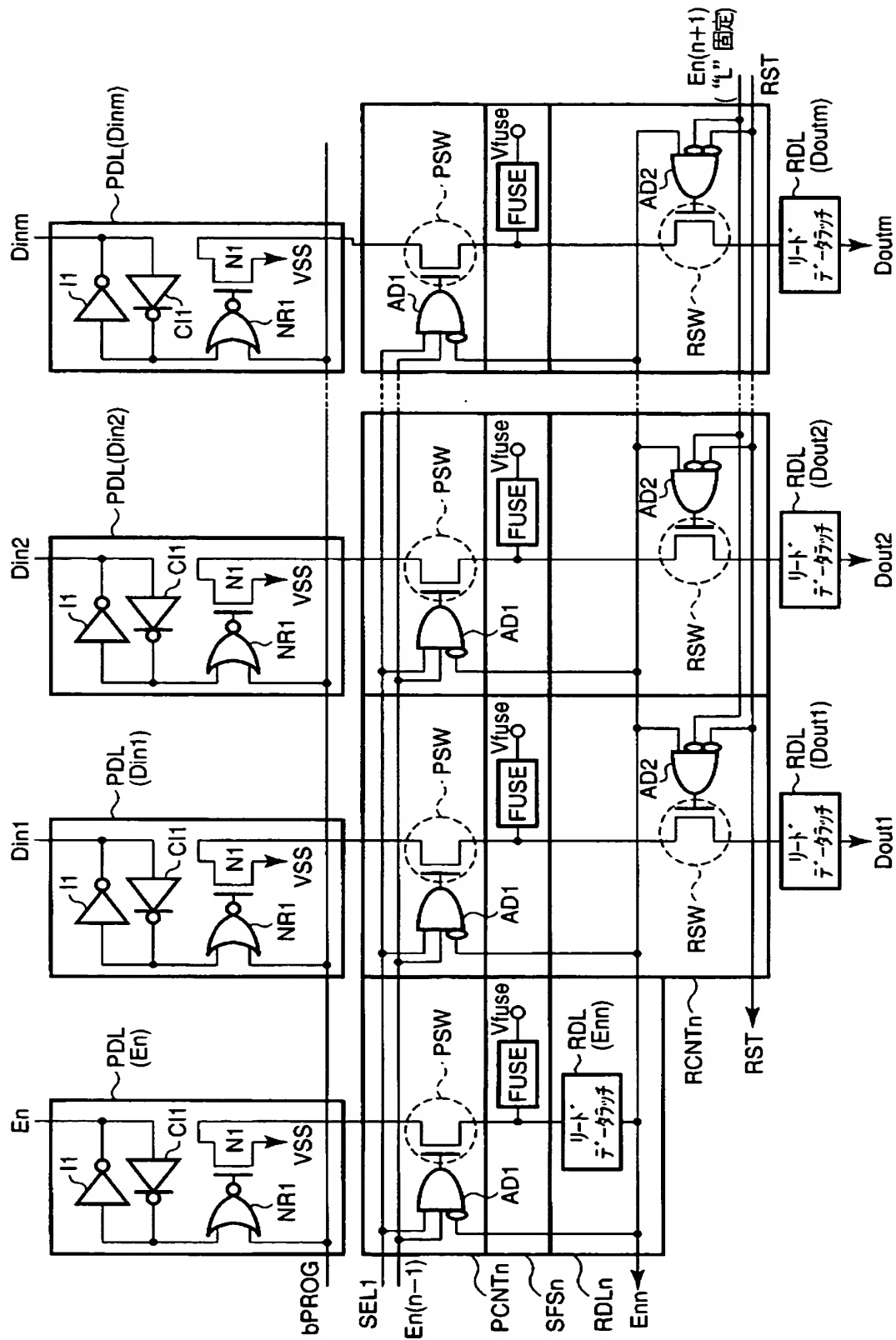
【図 6】



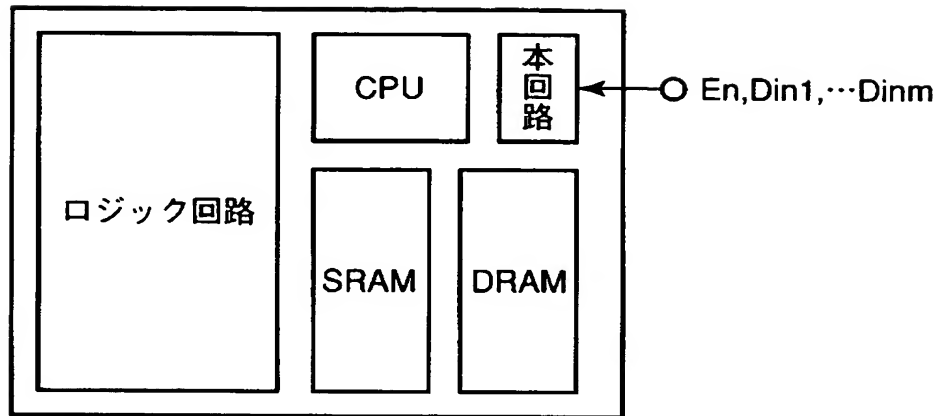
【図7】



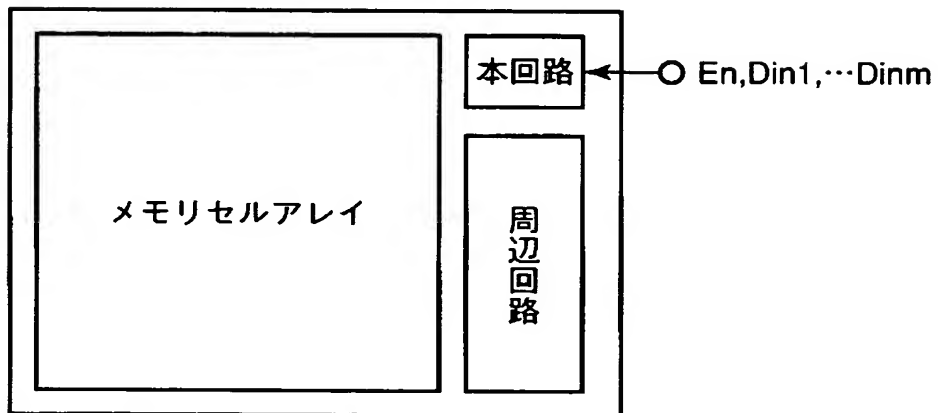
【図 10】



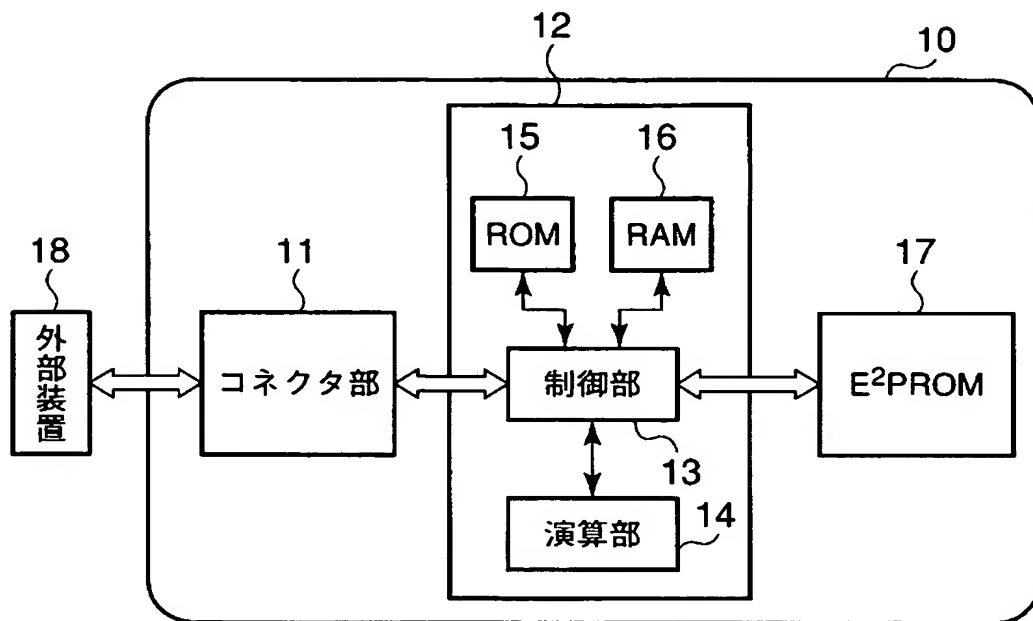
【図 1 1】



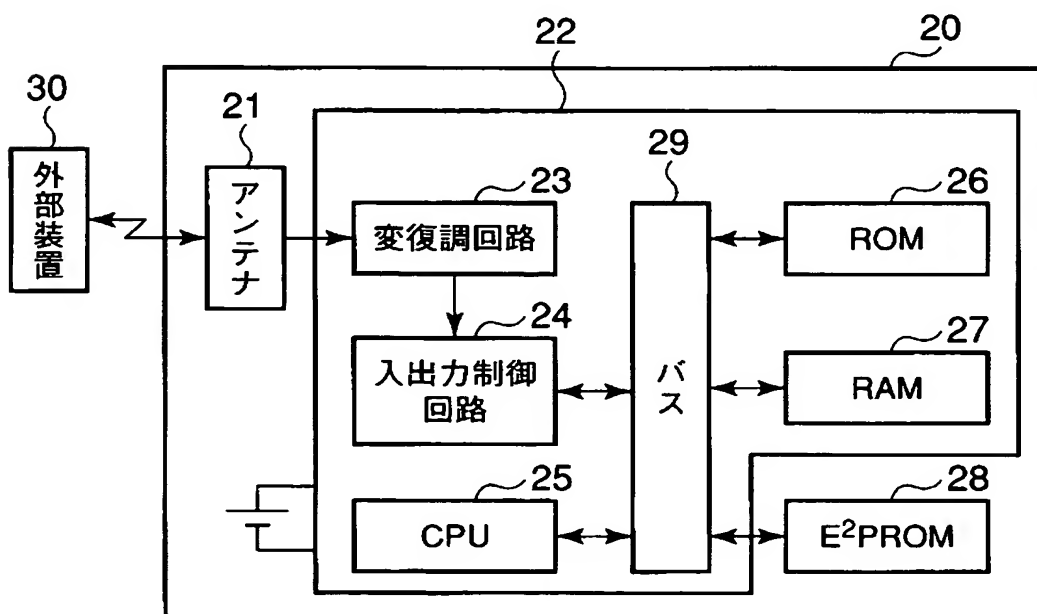
【図 1 2】



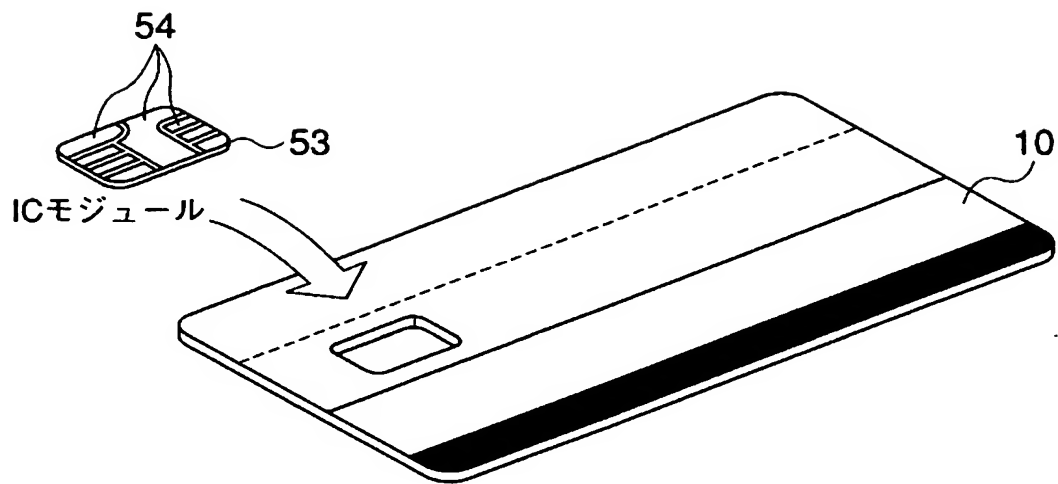
【図 13】



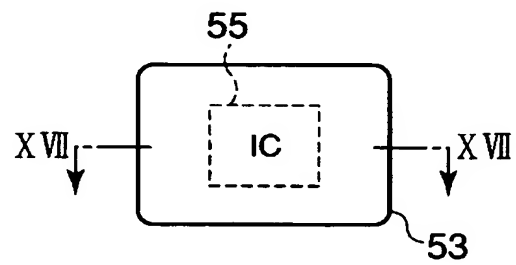
【図 14】



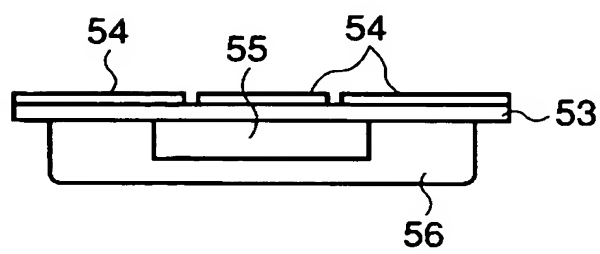
【図 15】



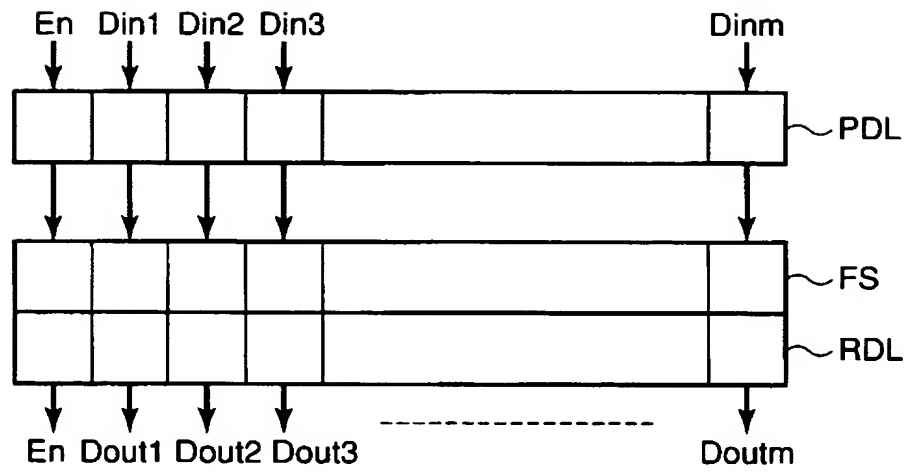
【図 16】



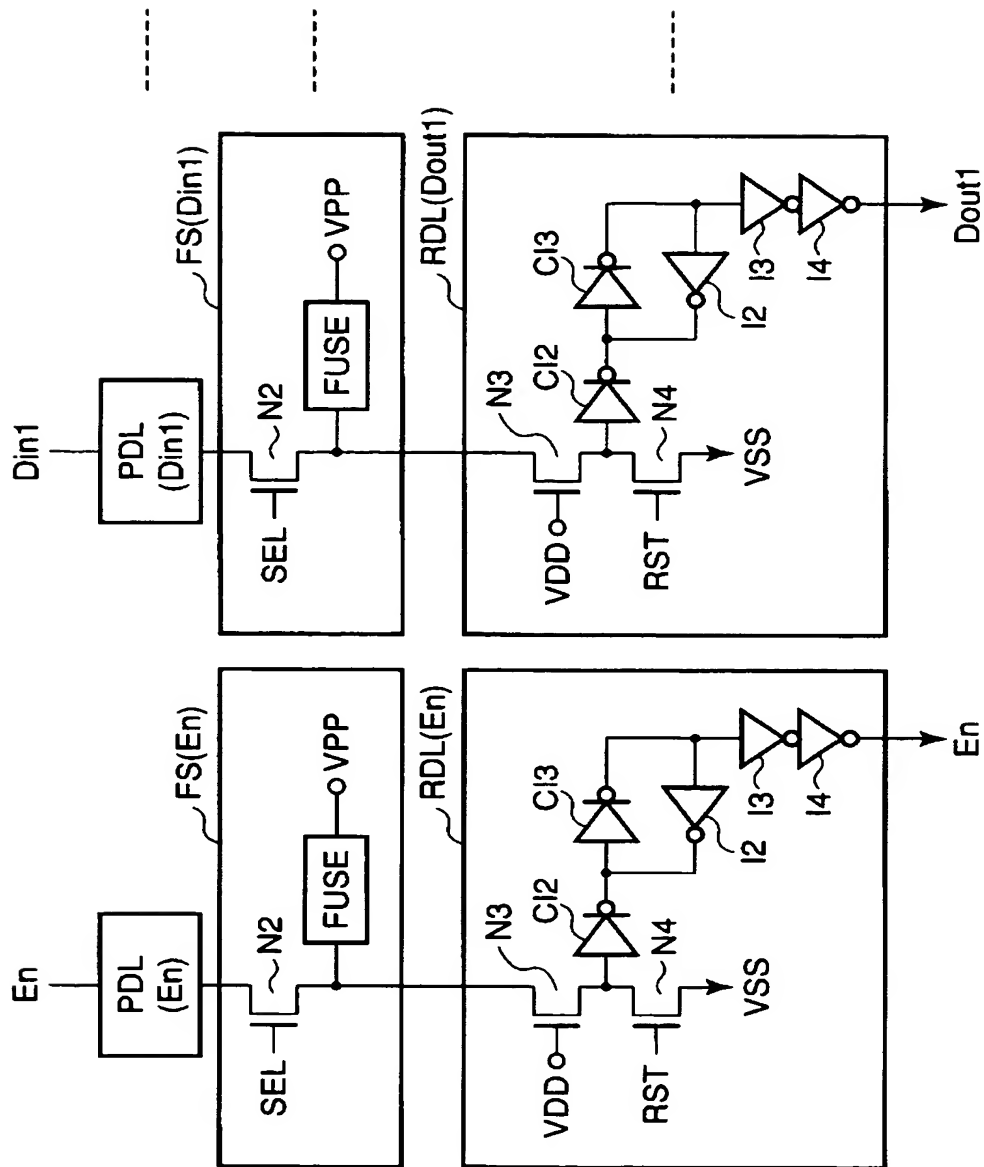
【図 17】



【図 1 8】



【図 19】



【書類名】 要約書**【要約】**

【課題】 複雑な制御なく、フューズセットに対するデータ書き換えを可能にする。

【解決手段】 1つのフューズセットは、複数のフューズブロック B L K 1, B L K 2, . . . B L K n から構成される。複数のフューズブロック B L K 1, B L K 2, . . . B L K n の各々は、電氣的にプログラム可能な複数のフューズ素子から構成されるサブフューズセット S F S 1, S F S 2, . . . S F S n と、複数のフューズ素子に対するプログラムを制御するプログラム制御回路 P C N T 1, P C N T 2, . . . P C N T n とを有する。複数のフューズ素子のうちの 1 つは、イネーブルビットであり、このイネーブルビットの値に基づいて、プログラムの対象となる 1 つのフューズブロックが決定される。

【選択図】 図 2

特願 2 0 0 3 - 2 9 0 8 6 1

出 願 人 履 歴 情 報

識別番号

[0 0 0 0 0 3 0 7 8]

- 1 . 変更年月日 1 9 9 0 年 8 月 2 2 日
 [変更理由] 新規登録
 住 所 神奈川県川崎市幸区堀川町 7 2 番地
 氏 名 株式会社東芝

- 2 . 変更年月日 2 0 0 1 年 7 月 2 日
 [変更理由] 住所変更
 住 所 東京都港区芝浦一丁目 1 番 1 号
 氏 名 株式会社東芝